

AD-755 393

**SLAMCODE: FINITE ELEMENT STRESS WAVE
ANALYSIS**

D. W. McCowan

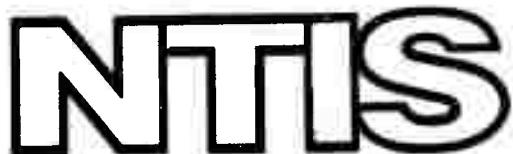
Teledyne Geotech

Prepared for:

Air Force Office of Scientific Research

14 September 1972

DISTRIBUTED BY:



National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

**BEST
AVAILABLE COPY**



*.....contributing to man's
understanding of the environment world*

31 OCT 1972

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

BY

D.W. McCOWAN

SPONSORED BY
ADVANCED RESEARCH PROJECTS AGENCY

MONITORED BY
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

ARPA ORDER NO. 1357

14 SEPTEMBER 1972



TELEDYNE GEOTECH

ALEXANDRIA LABORATORIES

Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U.S. Department of Commerce
Springfield VA 22151

Approved for public release;
distribution unlimited.

6D
R

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

| | |
|--|---|
| 1. ORIGINATING ACTIVITY (Corporate author) Teledyne Geotech Alexandria, Virginia | 2a. REPORT SECURITY CLASSIFICATION Unclassified 2c. GROUP |
|--|---|

3. REPORT TITLE

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (Last name, first name, initial)

McCowan, D.W.

| | | |
|-------------------------------------|------------------------------|----------------------|
| 6. REPORT DATE 14 September 1972 | 7a. TOTAL NO. OF PAGES 60 | 7d. NO. OF REFS 3 |
|-------------------------------------|------------------------------|----------------------|

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO. F44620-69-C-0082 | 8a. ORIGINATOR'S REPORT NUMBER(S) AL-72-1 |
| 8b. PROJECT NO. AO 1357 | |
| c. 62701D | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR - TR - 73 - 0120 |
| d. | |

10. AVAILABILITY/LIMITATION NOTICES

Approved for public release; distribution unlimited.

| | |
|--|---|
| 11. SUPPLEMENTARY NOTES TECH, OTHER | 12. SPONSORING MILITARY ACTIVITY Air Force Office of Scientific Res. 1400 Wilson Boulevard NPG Arlington, Virginia |
|--|---|

13 ABSTRACT

The program SLAMCODE developed in this study and described in this Volume is a large Finite Element Method (FEM) code capable of dynamic analysis of transient problems. It was adapted from the original SLAMCODE written by C. J. Costantino (Costantino, 1968) by incorporating the implicit Newmark B-integration (Newmark, 1959) and by constraining the free surface to be stress free. The present code also differs from the original by not permitting plastic elements. It calculates solutions for axisymmetric, plane stress or plane strain problems, as in the original code. It is designed to run on the IBM 360 family of computers under OS/360. Being written in Fortran, it can be easily adapted to other computer systems. In fact, it was written with this requirement in mind by minimizing the number of system-dependent Fortran peculiarities.

14. KEY WORDS

Finite Element
Stress Waves

1

Unclassified

Security Classification

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

ALEXANDRIA LABORATORIES REPORT NO. AL-72-1

| | |
|-----------------------------|--|
| Effective Date of Contract: | 1 March 1969 |
| Contract Expiration Date: | 31 March 1972 |
| Amount of Contract Dollars: | \$ 590,572 |
| Program Code: | 1F10 |
| Contract Number: | F-44620-69-C-0082 |
| ARPA Order No.: | 1357 |
| Principal Investigator: | Carl A. Newton |
| Period Covered: | 1 March 1969 through 31 March 1972 |



Approved for public release;
distribution unlimited.

TABLE OF CONTENTS

| | Page No. |
|---|----------|
| SLAMCODE Computer Program | 1 |
| REFERENCES | 5 |
| APPENDICES | |
| APPENDIX A | |
| SLAMCODE SOURCE LISTING | |
| APPENDIX A1 | |
| (MINBND) SOURCE LISTING | |
| APPENDIX B | |
| ØS/360 SLAMCODE LINK EDITOR CONTROL CARDS | |
| APPENDIX C | |
| ØS/360 FORTRAN H SLAMCODE JOB CONTROL LANGUAGE CARDS | |
| a. Requires two 9-track tape drives | |
| b. Requires four 9-track tape drives | |

LIST OF TABLES

| Table Title | Table No. |
|---|-----------|
| Stress-Constrained SLAMCODE Parameter Limits | 1 |
| 360 Stress-Constrained SLAMCODE Run Chart | 2 |
| Card Input Chart for 360 Stress-Constrained SLAMCODE | 3 |
| Data Set Chart for 360 Stress-Constrained SLAMCODE | 4 |

FIGURE TITLE

Figure Title

Page No.

**SLAMCODE overlay structure. Brackets
indicate subroutines.**

2

5

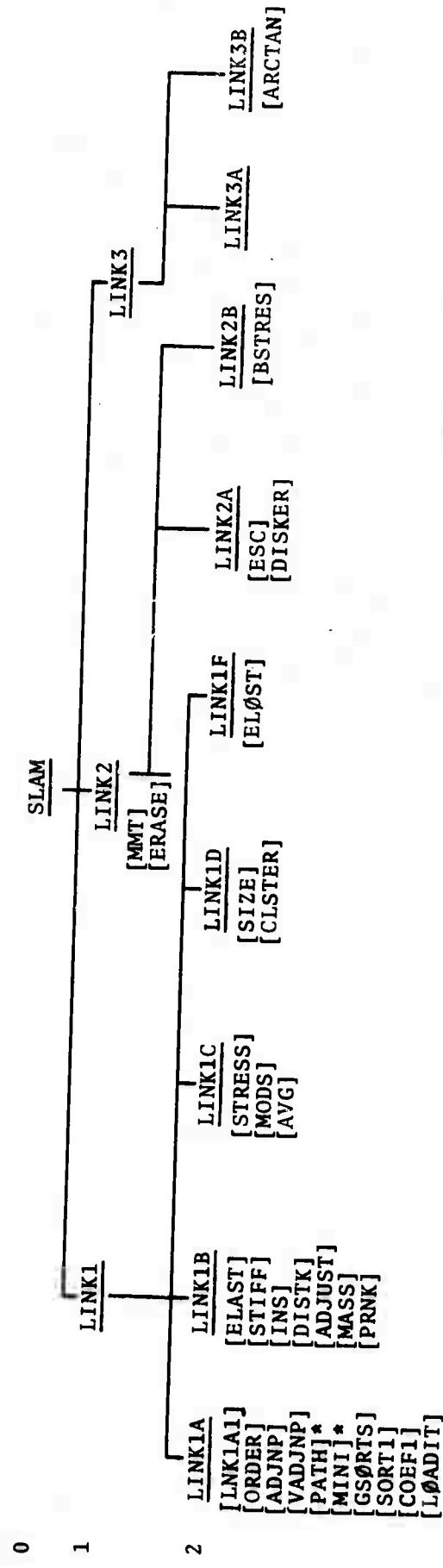
SLAMCODE Computer Program

The program SLAMCODE developed in this study and described in this Volume is a large Finite Element Method (FEM) code capable of dynamic analysis of transient problems. It was adapted from the original SLAMCODE written by C. J. Costantino (Costantino, 1968) by incorporating the implicit Newmark β -integration (Newmark, 1959) and by constraining the free surface to be stress free. The present code also differs from the original by not permitting plastic elements. It calculates solutions for axisymmetric, plane stress or plane strain problems, as in the original code. It is designed to run on the IBM 360 family of computers under OS/360. Being written in Fortran, it can be easily adapted to other computer systems. In fact, it was written with this requirement in mind by minimizing the number of system-dependent Fortran peculiarities.

The Fortran source cards are listed in Appendix A. The Link Editor control cards are listed in Appendix B. These control cards are appended to the object module for execution of the program. Appendix C contains examples of Job Control Language cards for the Fortran H computer using either two or four 9-track tape drives. Tables 1-4 describe the operation of the program and Figure 1 is a block diagram of the overlay structure. Finally, detailed descriptions of the input parameters and operation of the program can be gleaned by reading the comments in the source listing. All input data is described

Level 1

7



*[PATH] and [MINI] may be replaced by Rosen's matrix bandwidth minimization.

Figure 1. SLAMCODE overlay structure. Brackets indicate subroutines.

where it is read in the program listing.

The bandwidths of the diagonal matrices are controlled by the numbering of the nodal points. The bandwidth minimization algorithms listed in Appendix A are those used by Costantino, (PATH) and (MINI). A third algorithm written by Rosen (1968) and adapted for SLAMCØDE is listed in Appendix A1, (MINBND). The listing of the original Rosen algorithm contained some typographical errors which were corrected for (MINBND), but the comment cards which we have not reproduced in Appendix A1 are very useful. (PATH) is much faster than (MINBND) but its effectiveness depends upon the selection of starting node points. (MINI) has not decreased the bandwidth for any of our element arrays.

In general, large problems are solved by several runs of the program, all output being saved on magnetic tape or cataloged disk files. A wide range in input/output (I/O) flexibility is available with ØS/360 and this should be exploited to maximize program efficiency. For example, tape units should only be used in preference to disks when the time reduction made possible by saving intermediate results exceeds the time lost by the tape drives.

The program in its present form is I/O bound, i.e., a large amount of the computer time is spent spinning tapes and disks. In a typical 360/67 run there are approximately the same number of I/O seconds as there are CPU seconds. This performance

might be improved on larger machines where I/O buffers and program dimensions could be increased. The 360/67 version runs in a 280K byte region of core memory.

The number of I/O devices required depends on the operating system. The 360/67 system can assign many files to one device. A general rule of thumb for a system with 280K bytes of available core memory is that the program requires at least four tape drives and a hundred cylinders of disk file space.

REFERENCES

Costantino, C. J., 1968, Stress Waves in Layered Arbitrary Media, Final Report to Space and Missile Systems Organization (SAMSO) Norton Air Force Base, California: SAMSO TR 68-181.

Newmark, M., 1959, A Method of Computation for Structural Dynamics, J. of Engineering Mechanics Division, ASCE, 85, EM 3, 67-94.

Rosen, R., 1968, Matrix Bandwidth Minimization, Proceedings of 23d National Conference, ACM, Brandon/Systems Press, Inc., New Jersey, pp 585-595.

10

Table 1
STRESS-CONSTRAINED SLAMCODE PARAMETER LIMITS

| PARAMETER | DESCRIPTION | MAXIMUM VALUE | COMMENT |
|-----------|---|---------------|---|
| NUMNP | NUMBER OF NODE POINTS | 1000 | |
| NUMEL | NUMBER OF ELEMENTS | 1250 | NEVER ACTUALLY SET BUT IMPLIED IN A DIMENSION STATEMENT |
| -- | NUMBER OF ADJACENT NODES TO ANY NODE POINT | 8 | ≤ 6 FOR STRESSED NODES |
| LNES | NUMBER OF STRESSED LINES | 1 | |
| LADNP | NUMBER OF NODES PER STRESSED LINE | 50 | |
| NZONES | NUMBER OF MATERIAL ZONES | 20 | |
| NUMST | NUMBER OF STARTING NODES | 20 | |
| -- | MATRIX BLOCK SIZE=MESH BANDWIDTH+1 NODES | 30 | LINK1 LIMIT ON BANDWIDTH IS 100; HOWEVER, LINK2 LIMIT IS 29 |
| -- | PLASTIC ELEMENT TAPE BLOCKING FACTOR | 26 | NOT A CONSIDERATION IN THIS VERSION OF SLAM |
| -- | NUMBER OF GRAND PARTITIONS IN PATH PROCEDURE | 100 | PARAMETER INVOLVED IN BANDWIDTH MINIMIZATION |
| -- | NUMBER OF BLOCKS OF NODES ON STIFF./STRESS TAPE | 100 | e.g., MINIMUM BANDWIDTH FOR 1000 NODE PROBLEM = 9 |
| NUMOUT | NUMBER OF OUTPUT NODES | 100 | |

Table 2
360 STRESS-CONSTRAINED SLAUGHTER RUN CHART

| KRUN | CARD GROUPS SKIPPED | OUTPUT SKIPPED | TAPE ASSIGNMENTS | LINK 2 RUN TIME | LINK 3 RUN TIME | DESCRIPTION & COMMENTS |
|------|--|--|---|-----------------|-----------------|---|
| 0 | NONE | NONE | STIFF. & STRESS TAPE OUT ON #10 DTR TAPE OUT ON #20 IF TMAX #0 OTH TAPE OUT ON #8 IF TMAX #0 OTH TAPE OUT ON #3 IF TMAX #0 | 0 → TMAX | TSTRESS → MAX | INITIAL RUN THERIN MUST BE 0 MESH TITLE WRITTEN ON STIFF & STRESS, DTR, OTH, & OTH TAPES |
| 1 | NONE | ALL LINK1 STIFF. & MASS & STRESS TABLES | STIFF. & STRESS TAPE IN ON #10 DTR TAPE OUT ON #20 OTH TAPE OUT ON #8 OTH TAPE OUT ON #3 IF TMAX #0 | 0 → TMAX | TSTRESS → TMAX | LINK1 RERUN THERIN MUST BE 0 PROBLIM TITLE WRITTEN ON UTH, DTH, & OTH TAPES |
| 2 | ALL LINK1 INPUT 1.1 - 1.6 | ALL LINK1 OUTPUT | STIFF. & STRESS TAPE IN ON #10 DTR TAPE IN ON #20 OTH TAPE IN/OUT ON #8 OTH TAPE OUT ON #3 | THERIN+TMAX | TSTHSS+TMAX | LINK 2 MERUN |
| 3 | ALL LINK1 INPUT 1.1 - 1.6 | ALL LINK1 OUTPUT | STIFF. & STRESS TAPE IN ON #10 DTR TAPE IN ON #20 OTH TAPE IN/OUT #8 OTH TAPE OUT ON #3 | THERIN → TMAX | TSTRESS → TMAX | LINK2 SUMMARY OUTPUT SUMMARY=NO LINK3 STRESS OUTPUT |
| 4 | ALL LINK1 INPUT ALL LINK2 INPUT 1.1-2.2 | ALL LINK1 OUTPUT ALL LINK2 OUT- PUT STRESS OUT- PUT | STIFF. & STRESS TAPE IN ON #10 DTR TAPE IN ON #8 OTH TAPE OUT ON #3 | NOT RUN | TSTRESS → TMAX | LINK3 SUMMARY OUTPUT |
| 5 | ALL LINK1 INPUT ALL LINK2 INPUT 1.1-2.2 | ALL LINK1 OUTPUT ALL LINK2 OUT- PUT | STIFF. & STRESS TAPE IN ON #10 DTR TAPE IN ON #8 OTH TAPE OUT ON #3 | NOT RUN | TSTRESS → TMAX | LINK3 RERUN |

Table 3
CARD INPUT CHART FOR 360 STRESS-CONSTRAINED SLAMCODE

| CARD GROUP | READING ROUTINE | CONDX. OF READ | READ LIST | FORMAT | PARAMETERS |
|------------|-----------------|----------------|--|--------------------------------|---|
| 0.1 | SLAM | | KRUN,ANAME | (15,9A8) | KRUN ANAME - RUN TYPE SWITCH - 72 CHAR.RUN TYPE IDENTIFIER |
| 1.1 | LNK1A1 | | TITLE NUMNP,NUMEL,ISTRES,IPRINT | (10A8) (4I5) | TITLE NUMNP NUMEL ISTRES IPRINT - MESH TITLE/PROBLEM TITLE - # NODES - # ELEMENTS - GEOMETRY SWITCH - PRINT SWITCH |
| 1.2 | LNK1A1 | NPN < NUMNP | ANAME NPN,R(NPN),Z(NPN),ITYPE(NPN),THETA(NPN) | (10A8) (15,2E10.4,E10.4) | ANAME NPN R - CARD GROUP IDENTIFIER (NOT PRINTED) - NODE # - RADIUS - DEPTH - NODE TYPE - ROLLER ANGLE |
| 1.3 | LNK1A1 | | ANAME LINES,(LØADNP(I),I=1,LINES) (NPLØAD(I,J),SNØRM(I,J),J=1,LØADNP(I)) | (10A8) (14I5) (15,E10.2) | ANAME LINES LØADNP(I) NPLØAD(I,J) SNØRM(I,J) - LADING TYPE IDENTIFIER - STRESSED LINES - STRESSED LINES ON LINE I - MODE# OF JTH NODE ON STRESSED LINE I - SURFACE NORMAL ANGLE |
| 1.4 | LNK1A1 | I<LINES | ANAME NZONES IZ,ANAMF | (10A8) (15) (15,9A8) | ANAME NZONES IZ ANAME IELAST IPLAST WGT E1 E2 E3,E4,E5 - CARD GROUP IDENTIFIER (NOT PRINTED) - MATERIAL ZONES - ZONE # - ZONE MATERIAL IDENTIFIER - ELASTICITY SWITCH - PLASTICITY SWITCH - DENSITY - ELASTIC MODULUS - POISSON'S RATIO - ANISOTROPIC PARAMETERS |
| 1.5 | LNK1A1 | NUME<NUME | ANAME NUME,IZONE,NPI,NPJ,NPK,NPL | (10A8) (6I5) | ANAME NUME IZONE NPI-NPL - CARD GROUP IDENTIFIER (NOT PRINTED) - ELEMENT # - ZONE # - NODE # OF WHOLE ZONE |

Table 3 (Cont'd.)

| | | | | |
|-----|--------|---|--------------------------|---|
| 1.6 | LNK1A1 | ANAME NUMST (NSTART(1), I=1, NUMST) | (10A8) (15) (1415) | A NAME - CARD GROUP IDENTIFIER (NOT PRINTED) NUMST - # STARTING NODES NSTART - STARTING NODE #'S |
| 2.1 | L_NK2 | TMAX, TRERUN, ET, KDT, KINT, BETA, PDAMP | (3E10.0, 215, 2E10.0) | TMAX - MAX. RUN TIME FOR PRESENT RUN TRERUN - RESTART TIME ET - TIME INCREMENT KDT - TIME INCREMENT SWITCH KINT - DIVISOR FOR CHOOSING DT BETA - INTERPOLATION SCHEME PARAMETER PDAMP - CRITICAL DAMPING FOR ARTIFICIAL VISCOSITY |
| 2.2 | BSTRES | IPULSE, IDIREC, N1, N2, STAMP | (415, E10.2) | IPULSE - PULSE IDIREC - DIRECTION SWITCH N1 - FIRST STRESSED NODE N2 - LAST STRESSED NODE STAMP - STRESS AMPLITUDE |
| 3.1 | LINK3A | TSTRSS, TMAX, NUROUT, IØUT, JØUT | (2E10.0, 315) | TSTRSS - OUTPUT START TIME TMAX - OUTPUT STOP TIME NUROUT - # OUTPUT NODES IØUT - # INTEGRATION POINTS/PRINTED OUTPUT JØUT - # INTEGRATION POINTS/TAPE OUTPUT NPØUT - OUTPUT NODES #'S |
| | | (NPØUT(1), I-1, NUMØUT) | (1415) | |

Table 4
DATA SET CHART FOR 360 STRESS-CONSTRAINED SLAMCODE

| PROGRAM | DSRN | I/O | DEVICE | SAVED | COMMENT |
|---------|------|-----|---|--------------------------------|--|
| LINKIA | 1 | 0 | SORTED ELEMENT DATA | DISK | NEVER |
| | 4 | 0 | SORTED NODE DATA | DISK | NEVER |
| | 8 | 0 | ADJACENCY TABLES | DISK | NEVER |
| | 14 | 0 | PLASTIC ELEMENT DATA | DUMMY | NEVER USED IN THIS VERSION |
| LINKIB | 1 | 1 | AS ABCVE UNBLOCKED STIFFNESS TABLES | DISK | NEVER |
| | 3 | 0 | AS ABOVE | LINK1 SCRATCH | |
| | 4 | 1 | AS ABOVE | LINK1 SCRATCH | |
| | 8 | 1 | AS ABOVE | LINK1 SCRATCH | |
| | 12 | 0 | LINKIC ELEMENT DATA | DISK | NEVER |
| LINKIC | 3 | 0 | UNBLOCKED STIFFNESS AND STRESS TABLES | DISK | NEVER |
| | 4 | 1 | AS ABOVE | CONTINUATION OF DATA SET ABOVE | |
| | 8 | 1 | AS ABOVE | | |
| | 12 | 1 | AS ABOVE | | |
| LINKID | 3 | 1 | AS ABOVE BLOCKED STIFFNESS AND STRESS TAPE | TAPE | AFTER KRUN=0 RUN |
| | 10 | 10 | PLASTIC SAVE TAPE | TAPE | OUTPUT FOR KRUN=0, INPUT FOR KRUN=1 |
| LINKIF | 12 | 0 | PLASTIC SAVE TAPE | TAPE | THIS VERSION OF SLAMCODE IS INCAPABLE OF PLASTIC ANALYSIS; HENCE, LINKIF SHOULD NOT BE EXECUTED. |
| LINK2 | 8 | 10 | DISPLACEMENT TIME HISTORY TAPE | TAPE | AFTER KRUN=0 (TMAX#0) |
| | | | | | WRITES TITLE AND COMMON BLOCK RECORDS FOR KRUN=0, ¹ READS TITLE AND COMMON BLOCK RECORDS FOR KRUN=2, ³ |
| LINK2A | 10 | 1 | AS ABOVE UPPER TRIANGULARIZATION TAPE | TAPE | AFTER KRUN=0 (TMAX#0) AND KRUN=1 RUNS |
| | 20 | 0 | NON-SEQUENTIAL SCRATCH | DISK | NEVER |
| LINK2B | 8 | 10 | AS ABOVE | TAPE | AFTER KRUN=0 (TMAX#0) |
| | | | | | WRITES PROBLEM INTEGRATION OUTPUT FOR KRUN=0,1, ² , ³ CONTINUATION OF DATA SET ABOVE INPUT FOR KRUN=2, ³ |
| LINK3 | 8 | 1 | AS ABOVE | | READS TITLE AND COMMON BLOCK RECORDS FOR ALL KRUN |
| LINK3A | 10 | 1 | AS ABOVE | | READS BLOCKED STRESS TABLES FOR ALL KRUN |
| LINK3B | 3 | 0 | OUTPUT TIME HISTORY TAPE AS ABOVE | TAPE | AFTER ALL KRUN |
| | 8 | 1 | | | PROBLEM OUTPUT FOR ALL KRUN READS INTEGRATION OUTPUT FOR ALL KRUN |

APPENDIX A. SLANCODE SOURCE LISTING

```

C PROGRAM SLAM
C IMPLICIT REAL*(A-H,O-Z)
C COMMON MAXNP,MAXJP,MXLINP,MXPMP,MAXELB,NUNNP,NUMEL,
1  TSTRS,LINES,NUMPFL,LOADNP(1),PERIOD,DT,NLOAD(1,50),
2  RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXRD,
3  MAXLOC,NREADS,MAXMKR,TRFL,POAMP,RETA,KRUN,
4  MXZONE,NZONE,S,PRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
5  DIMENSION NPTN(1000),IELAST(20),F1(20),F2(20),F3(20),
1E4(20),E5(20)
6  DIMENSION ANAME(91)

C ITEM 360/67 VERSION OF VISCO-ELASTIC SLAM CODE
C CONVENTED APRIL, 1971 BY D.W. MCCORMAN WITH CONSULTATION FROM
C J. CONSTANTINO
C
C UPDATED TO DO IMPLICIT TIME INTEGRATION OCTOBER, 1971
C UPDATED TO CONSTRAIN AROUNDARY STRESS MARCH, 1972
C
C SET LIMITS
C
C CALL NOIFY
C MAXNP=1000
C MAXJP=A
C MLINE=1
C MLOAD=50
C MNPD=100
C MAXELB=26
C MXZONE=2D
C MAXRK=30

MAXNP = MAXIMUM NUMBER OF NODE POINTS
MAXJP = MAXIMUM NUMBER OF NODE POINTS ADJACENT TO ANY NODE POINT
MLINE = MAXIMUM NUMBER OF LOADED LINES OR SURFACES
MLOAD = MAXIMUM NUMBER OF LOADED NODES PER LOADED LINE
MNPD = MAXIMUM NUMBER OF NODE POINTS PER BUFFER
MAXELB = MAXIMUM PLASTIC ELEMENTS PER BLOCK
MXZONE = MAXIMUM NUMBER OF MATERIAL ZONES
MAXRK = MAXIMUM BANONIOTH FOR INTEGRATION SCHEME

READ(5,1) KRUN,ANAME
1 FORMAT(15,90)
WRITE(6,4) KRUN,ANAME
4 FORMAT(1M1,SMRUN=,1$IX,900)

KRUN = SWITCH FOR RERUN
KRUN = 0 INITIAL RUN
      = 1 LINK1 RERUN
      = 2 LINK2 RERUN
      = 3 LINK3 SUMMARY RERUN
      = 4 LINK4 SUMMARY RERUN
      = 5 LINK5 RERUN
ANAME = DESCRIPTOR ON DATA CARD, FOR RUN TYPE ONLY

```

```

IF(KRUN,GT,1) GO TO 3
CALL LINK1
3 IF(KRUN,GT,3) GO TO 5
CALL LINK2
5 CONTINUE

SUBROUTINE LINK1
IMPLICIT REAL*(A-H,O-Z)
COMMON MAXNP,MAXJP,MXLINP,MXPMP,MAXELB,NUNNP,NUMEL,
1  TSTRS,LINES,NUMPFL,LOADNP(1),PERIOD,DT,NLOAD(1,50),
2  RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXRD,
3  MBLCK,NREADS,MAXMKR,TRFL,POAMP,RETA,KRUN,
4  MXZONE,NZONE,S,PRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
DIMENSION NPTN(1000),IELAST(20),WGTL(20),E1(20),E2(20),F3(20)
1E4(20),E5(20)

C LINK1 PREPARES THE STIFFNESS, MASS, STRESS, AND PLASTIC TABLE
C FOR THE RUN
C
C CALL LINK1A
IF(KRUN,EQ,1) GO TO 10
CALL LINK1A
CALL LINK1C
10 CALL LINK1B
IF(KRUN,EQ,1,OR,NUMPFL,FO,0) RETURN
CALL LINK1F
RETURN
END

SUBROUTINE LINK1A
IMPLICIT REAL*(A-H,O-Z)
COMMON MAXNP,MAXJP,MXLINP,MXPMP,MAXELB,NUNNP,NUMEL,
1  TSTRS,LINES,NUMPFL,LOADNP(1),PERIOD,DT,NLOAD(1,50),
2  RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXRD,
3  MBLCK,NREADS,MAXMKR,TRFL,POAMP,RETA,KRUN,
4  MXZONE,NZONE,S,PRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
DIMENSION NPTN(1000),RI,NAQNP(1000),NAQEL(1000),
1NPTP(1000),S(1000),NSTART(20),IELAST(20),IPLAST(20),
2WGTL(20),E1(20),E2(20),E3(20),E4(20),E5(20),R(1000),Z(1000),
3ITYPE(100),THETA(1000),TMP(5000),NTMP(8,1250)
DIMENSION NPTN(1000),
EQUivalence (TMP(1),NTMP(1,1))

C THIS LINK READS THE MESH DATA CARDS. RFNUMBERS THE MFSH.
C SORTS THE MESH DATA, AND PREPARES TABLES FOR THE STIFFNESS, STRESS, AND PLASTIC MATRICES CALCULATIONS
C SET MAXIMUM NUMBER OF STARTING NODES
C
C MKSTRT=20

```

```

C READ AND STORE CARD INPUT
C
C CALL LNKIA(1,MAXNP,NKL,INE,MXLOAD,MKZONE,NUMNP,NUMEL,(PRINT,
C 1ISTRES,NUMER,NZONES,R,2,I TYPE,TMFTALLNFS,LOADNP,NPLDOA,
C 2RAD2AO,SNDRM,TELAST,IPLAST,WGT,E1,E2,E3,F4,ER,NUMST,INST,
C 3INST,NTMP,TITLE)
C
C ZERO OUT EVERYTHING
C
C 00 1 I=1,NUMNP
C NAQJNP(I)=0
C NAQFEL(I)=0
C 00 S J=1,MAXQJP
C 5 NPADJ(I,J)=0
C
C CALCULATE NPADJ, NACUNP, NAQJEL
C
C NPADJ = ADJACENT NODE POINT NUMBERS ARRAY
C NAQJNP = NUMBER OF ADJACENT NODE POINTS ARRAY
C NAQJEL = NUMBER OF ADJACENT ELEMENTS ARRAY
C
C 00 7 M=1,NUMEL
C CALL ADNP(MXADJP,MAXNP,NUMNP,NAQJNP,NAQJFL,NTMP(12,M),NTMP(5,M),
C INTPP(6,M),NTMP(7,M),NTAP(8,M))
C CALL VADNP(MXADJP,NAOJNP,MAXNP,NUMNP,NAQJNP,NAQJEL)
C PRINT
C
C IF((PRINT,NE,1).AND.((PRINT,NE,0))) GO TO 4
C WRITE(6,1)
C 1 FORMAT(1H1,3BHTABLE OF ORIGINAL ADJACENT NODE POINTS//,
C 1 AX,4HNODE,13X,6HNO. OF,27X,20HADJACENT NODE POINTS//,
C 2 4X,5HPOINT,1IX,9HADJ. PTS.,1X,3HADJ. ELS.,5X,1H2,9X,1H2,9X,(H3,
C 3 9X,1H4,9X,1H5,9X,1H9,9X,1H7,9X,1H8//)
C 00 2 I=1,NUMNP
C 2 WRITE(6,3) I,NAQJNP(I),NAQJEL(I),(NPADJ(I,J),J=1,MAXQJP)
C 3 FORMAT(1B,8X,2I10,B10)
C
C CALCULATE ORIGINAL BANDWIDTH
C
C 4 CCNTINUF
C MAXBD=0
C 00 6 I=1,NUMNP
C NUM=NAQJNP(I)
C 00 6 J=1,NUM
C NP=NPADJ(I,J)
C NBNM=NBNS(I,NP-1)
C (FINBAN,LE,MAXBD) GO TO 6
C CCNTINUF
C
C 6 CCNTINUF
C RENUMERFR WITH PATH PROCFCNURF
C CALL PATH(MAXNP,NUMNP,NUMST,INST,INST,NTSTART,NTPT,MAXNP,NAQJNP,NAQJEL,
C 1IGP,NUMGP)
C
C NPTP = NEW NODE POINT NUMBERS IN OLD NODE POINT ORDER
C NPTN = OLD NODE POINT NUMBERS IN NEW NODE POINT ORDER
C PRINT
C
C WRITE(6,B) NUMGP
C 6 FORMAT(1H1,24HNO. OF GRAND PARTITIONS=,I5)
C 6 WRITE(6,9) ((IGP(I),I=1,NUMGP)
C 2 FORMAT(1,23H PARTITION NEW NODE NO./12X,1S,10X,(5))
C 450 WRITE(6,10) MAXBD
C 10 FORMAT(1H1/20H ORIGINAL BANDWIDTH=,I5)
C
C CALCULATE AND PRINT NEW BANDWIDTH
C
C MAXBD=0
C DO 11 I=1,NUMNP
C NPNP=NPTP(I)
C NUM=NAQJNP(I)
C DD 11 J=1,NUM
C NP=NPADJ(I,J)
C NPNW=NPTP(NP)
C NW=ABS(NPNP-NPNW)
C IF(NBAN,LE,MAXBD) GO TO 11
C MAXBD=NBAN
C 11 CONTINUE
C
C WRITE(6,12) MAXBD
C 12 FORMAT(20H NEW BANDWIDTH=,I5)
C
C MINIMIZE BANDWIDTH WITH MINIMAX PROCEOURF AND CHECK
C
C CALL MINIMAXP,NUMNP,NAQJNP,MXADJP,NPADJ,NPTP,NPTP,S,
C 1MAXBD)
C (FINBAN,LT,MXNPB) GO TO 14
C WRITE(6,13)
C 13 FORMAT(1H1/20H BANDWIDTH TOO LARGE)
C CALL EXIT
C
C PENUMBER NPAQJ
C
C 14 DD 15 I=1,NUMNP
C KN=NACINP(I)
C DO 15 J=1,KN
C KT=NPADJ(I,J)
C 15 NPAQJ(I,J)=NPTP(KT)
C PRINT
C
C 16 FORMAT(1H1,3BHTABLE OF NEW ADJACENT NODE POINTS//,
C 14X,4MNEN *3X,4HOLD *6X,6HNO. OF,4X,6HNO. OF,27X,
C 22HADJACENT NODE POINTS/4X,4HNODE,3X,4HNODE,5X,1H2,9X,1H3,
C 31X,9HACJ. FLS.,5X,1H1,9X,1H2,9X,1H3,
C 4 9X,1H4,9X,1H5,9X,(H9,9X,1H7,9X,1H8//)
C DD 17 I=1,NUMNP
C
C (FINBAN,LE,MAXBD) GO TO 16
C WRITE(6,16)
C 16 FORMAT(1H1,3BHTABLE OF NEW ADJACENT NODE POINTS//,
C 14X,4MNEN *3X,4HOLD *6X,6HNO. OF,4X,6HNO. OF,27X,
C 22HADJACENT NODE POINTS/4X,4HNODE,3X,4HNODE,5X,1H2,9X,1H3,
C 31X,9HACJ. FLS.,5X,1H1,9X,1H2,9X,1H3,
C 4 9X,1H4,9X,1H5,9X,(H9,9X,1H7,9X,1H8//)

```

```

KP=NPTN(1)
17 WRITE(6,13) 1,KP,NADJNP(KP),NADJEL(KP),(NPADJ(KP,J),J=1,MXAQJP)
18 FORMAT(2I8,2I10,8I10)
19 GO TO 26
19 IF(IPRINT.EQ.0) GO TO 26
20 FORMAT(1H1,2I4 NEW NUMBERING SCHEME//)
DO 21 I=1,NUMNP,10
IF((NUMNP-1).LT.10) IOUN=NUMNP-1
IF((NUMNP-1).GE.10) IOUN=10
23 WRITE(6,24) I,J1,IUM1
24 FORMAT(1/20H ORIGINAL NODES ,10I10)
21 WRITE(6,25) IMTP(J1,J=1,IUM1)
25 FORMAT(20H NEW NODES ,10I10)
C WRITE ADJACENCY TABLES ON TAPE
C
C 26 CONTINUE
500 IF(IKRN.EQ.1) GO TO 501
REINC 8
DO 22 I=1,NUMNP
KP=NPTN(1)
22 WRITE(8) 1,NADJNP(KP),NADJEL(KP),(NPADJ(KP,J),J=1,MXAQJP)
REWIND 8
C
C RENUMBER NPMLOAD
C
C 501 DC 32 /*LINES
NUM=LOADNP(1)
DO 32 J=1,NUM
NPMLOAD=NPMLOAD(1,J)
NPMLOAD(1,J)=NPTP(NPMLOAD)
C
C REMEMBER ELEMENT VERTICES AND SET ELEMENT DATA ONTO TAPE 1 IN
C INCREASING LOWEST NEW NODE POINT ORDER
C
C REMEMBER COLUMNS OF NTMP ARE - KEY,NAME ,IZONE,KASe,NPj,NPK,NPl
C
C 501 IF(IKRN.EQ.1) GO TO 631
DO 27 I=1,NUMEL
NTMP(5,1)=NPTP(NTMP(5,1))
NTMP(6,1)=NPTP(NTMP(6,1))
NTMP(7,1)=NPTP(NTMP(7,1))
IF(NTMP(6,1).EQ.0) GO TO 29
NTMP(6,1)=NPTP(NTMP(8,1))
NTMP(1,1)=MIN(NTMP(5,1),NTMP(6,1),NTMP(7,1),NTMP(8,1))
GO TO 27
29 NTMP(1,1)=MIN(NTMP(5,1),NTMP(6,1),NTMP(7,1))
C
C 27 CONTINUE
CALL GSORTS(NTMP,NUMEL,8,1,1)
C
C SORT NODE DATA ONTO TAPE 4 IN INCREASING N... NODE POINT ORDER
C
DO 30 N=1,NUMNP
NTMP(1,N)=NPTP(N)
CALL LOACITIRIN(NTMP(2,N))
C

```



```

C FORM NUMBER OF ACJACENT NODE POINTS ARRAY
C
C DO 12 M=1,NUMNP
C DO 10 I=1,MXADJP
C J=1
C IF (INPACJ(M,I).EQ.0) GO TO 11
C 10 CONTINUE
C NADJNPIM)=MXAOJP
C DO TO 12
C 11 NADJNPIM)=J-1
C 12 CONTINUE
C RETURN
C END

SUBROUTINE PATHIMAXNP,NUMNP,NUMST,NSTART,NPTN,NPTP,MXADJP,
1 NADJNP,NADJ,IGP,NUMGP)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION NSTART(NUMST),NPTN(MAXNP),NPTP(MAXNP),NADJNP(MAXNP),
1 INPADJIMAXNP,MXAOJP,IGP(1)

C MINIMIZE BANDWIDTH WITH PATH PROCEDURE
C
C NSN = NUMBER OF STARTING NODES
C NSNR = STARTING NODE ARRAY
C IGP = GRAND PARTITION ARRAY
C NUMGP = NUMBER OF GRAND PARTITIONS
C
C KOUNT=0
C IN=1
C DO 1 I=1,NUMNP
C NPTN(I)=0
C 1 NPTPII)=0

C DO 2 I=1,NUMST
C NP=NSTART(I)
C NPTP(NP)=IN
C KOUNT=KOUNT+1
C 2 NPTN(KOUNT)=NP
C IGP(I)=KOUNT

C 4 DO 7 I=1,NUMNP
C IF (INPTP(I).NE.IN) GO TO 7
C NUM=NACINP(I)
C DO 3 J=1,NUM
C NP=NPADJ(I,J)
C IF (INPTP(NP).NE.0) GO TO 3
C NPTP(NP)=IN+1
C KOUNT=KOUNT+1
C NPTN(KOUNT)=NP
C IF (KOUNT .EQ. NUMNP) GO TO 5
C 3 CONTINUE
C 7 CONTINUE

C MAXBD=0
DO 14 I=1,NUMNP
NPOLD=NPTN(I)
NUM=NADINP(NPOLD)
DO 10 J=1,NUM
NADJ=NPADJ(NPOLD,J)
DO 11 K=1,NUMNP
KK=K
IF (NADJ.EQ.NPTN(K)) GO TO 15
11 CONTINUE
WRITE(6,100) I,NPOLD,NPTN(I),L=1,NUMNP
100 FORMAT(1H1,13HERROR IN MINI//10X,21I//10X,10I1C)
CALL EXIT
15 NPNEW=KK
NBPN=TABS(NPNEW-1)
IF (MAXBC.LT.NBPN) MAXBD=NPN
10 CCNTINUE
14 CCNTINUE
C
C 7 WRITE(6,6) MAXBD
C 6 FORMAT(20H NEW
C BANDWIDTH=,15)

```

```

C IF(MAX80P.LE.MAXBD) GO TO 12
C
DC 5 I=1,NUMNP
NPOLD=NPTN(1)
5 NPTP(NPOLD)=I
MAXBD=MAXBD
GO TO 3
C
12 DO 16 I=1,NUMNP
 16 NPTN=NPTP(I)
 16 NPTN*NPNEW)=I
  RETURN
END
C
SUBROUTINE GSORTS(IARRAY,NRCD$,$NWRD$,NKEY,IOUTAP)
IMPLICIT REAL*8(A-H,D-Z)
COMMON MAXNP,MXAUP,MALINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
 1 ISTRU$,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLLOAD(1,50),
 2 RAD(1,50),ZAC(1,50),SNORM(1,50),TITLE(1C),MAXBD,
 3 MBLOCK,NREADS,MXMBK,TREAL,PDAUP,BETA,KRUN,
 4 MX2DNE,M2DNE,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5,
 5 DIMENSION NPTN(1000),NADJNP(100),NADJEL(100,8),NADJEL(100),
 12(I0D),ITYPE(100),THETA(100),XMASS(100),
 13(I0D),CK(4,4),CK(8,8),AINT(2C),
 14(I0D),MGT(20),E1(20),E2(20),E3(20),E4(20),E5(20),
 15ADUM(I0D,8),SNPUM(100),SNPWW(I0D),SDAUU(10C,8),
 16ADUM(I0D,8),SDWM(100,8)
C
THIS LINK CALCULATES THE MASS AND STIFFNESS MATRICES FOR EACH
ELEMENT, MAKES THE ROLLER SUPPORT MODIFICATIONS, AND DISTRIBU-
THE ELEMENT STIFFNESSES TO NODE POINT STIFFNESSES
C
KEND = CONTROL CONSTANT TO DETERMINE END OF PROCESSING
C
NPOUT = FIRST NODE POINT MINUS ONE IN BUFFERS
C
NUMCP = NUMBER OF COMPLETED NODE POINTS IN BUFFERS
C
NUMPR = NUMBER OF NODE POINTS IN BUFFERS (COMPLETED OR INITI-
C
NPR = NUMBER OF UNCOMPLETED NODE POINTS IN BUFFERS
C
KX = WHERE TO START ERASING BUFFERS
C
INITIALIZE
C
REWIND IOUTAP
CALL SORT1(IARRAY,IARRAY,NRCD$,NWRD$,NKEY,Y,C)
DO 3 I=1,NRCD$
 3 WRITE(IOUTAP)(IARRAY(J,1),J=1,NWRD$)
REWIND IOUTAP
RETURN
END
C
SUBROUTINE SORT1(IARRAY,JARRAY,NRECDS,IWRDS,JWRDS,JFFY,ISWT)
IMPLICIT REAL*8(A-H,D-Z)
DIMENSION IARRAY(IWRDS,NRCD$),JARRAY(JWRDS,NRECDS)
C
GENERAL PURPOSE SORTER
C
LOGICAL CHECK
M=NRECDS-1
C
1 CHECK=.FALSE.
DO 6 I=1,2
 6 DO 2 J=1,M/2
 2 IF(IARRAY(IKEY,J).LT.E.IARRAY(IKEY,J+1)) GO TO 2
 2 DO 3 K=1,IWRDS
 3 ITEMP=IARRAY(K,J)
  IARRAY(K,J)=IARRAY(K,J+1)
 3 ITEMP(IK,J+1)=ITEMP
 3 IF(IISWT.FQ.0) GO TO 5
 5 DO 4 K=1,JWRDS
 4 JTEMP=JARRAY(K,J)
  JARRAY(K,J)=JARRAY(K,J+1)
 4 JTEMP(IK,J+1)=JTEMP
 5 CHECK=.TRUE.
 2 CONTINUE
 6 CONTINUE
C
24
C

```

```

C   2 IF(MUNMNP.LT.MXNMP) NUMNPB=NUMNP
C     IF(MUNMNP.GE.MXNMP) NUMNPB=MXNMP
C     DO 3 (=1,NUMNPB)
C       READ(8) NPM,NADJNP
C       READ(8) NPM,NADJNP()
C       READ(4) NPM,R((1),2((1),ITYPF((1),THETA((1))
C 3 CONTINUE
C
C   READ ANOTHER ELEMENT DATA RECORD
C
C   ICOUNT=0
C   READ(11) KEY,NUME,IZONE,KASF,NTJ,NTK,NTL
C   ICOUNT=ICOUNT+1
C   LNP=MAXD(NTJ,NTJ,NTK,NTL)
C
C   CHECK TO SEE IF LARGEST NODE POINT IN THIS ELEMENT FALLS OUTSIDE
C   BUFFER
C
C   IF((LNP-NPOUT).GT.MXNMPB) GO TO 100
C
C   PROCESS THIS ELEMENT
C
C   6 NP((=NTI-NPOUT
C     NPJ=NTJ-NPOUT
C     NPK=NTK-NPOUT
C     IF(NTL.EQ.0) NPL=0
C     IF(NTL.NE.0) NPL=NTL-NPOUT
C     S1=0.0
C     C1=0.0
C     IE=IE-LAST(IZONE)
C     A1=E1(IZONE)
C     A2=E2(IZONE)
C     A3=E3(IZONE)
C     A4=E4(IZONE)
C     A5=E5(IZONE)
C     RHO=WGT(IZONE)/(3B6.4*172B.)
C     CALL ELAST1E,ISTRES,A1,A2,A3,A4,A5,C,NUME)
C     CALL STIFFKASE,NPI,NPJ,NPK,NPL,NUME,MXNPB,ISTRES,C,R,Z,CK,AINT,
C     IS1,C1)
C     CALL ADJUST(MXNPB,CK,ITYPE,THETA,NADJEL,NPI,NPK,NPL)
C     CALL OISTKMXNPB,MXADJP,CK,SNPBU,SNPUM,SNPWN,SADUW,SADWU,
C     ISADWW,NPI,NPJ,NPK,NPL,NPADJ,NPOUT)
C     CALL MASS(MXNPB,RHO,R,Z,AINT,YMASS,S1,C1,NPI,NPK,NPL,ISTRES)
C     WRITE(12) KEY,NUME,IZONE,NTJ,NTK,NTL,((C((1,J),1#164),J=1,4),
C     1KASE,S1,C1
C
C   GO TO READ ANOTHER ELEMENT RECORD OR SET STOP SWITCH
C
C   IF(ICOUNT.LT.NUMEL) GO TO 4
C     KEND=1
C     KEY=NUMNP+1
C
C   SET-UP FOR RUMPING BUFFERS
C
C   100 NUMCP=KEY-1
C     NUMNPB=NUMCP-NPOUT

```

```

      IF(TYPE(K)=1)TYPE(L)
      THETAK=THETAIL(L)
      R(K)=R(L)
      Z(K)=Z(L)
      XMASS(K)=XMASS(L)
      SNUU(L)=SNUU(L)
      SNPW(K)=SNPW(L)
      SNPW(K)=SNPW(L)
      00 903 J=1,MAXJ(L,J)
      NPAD(J,K,J)=NPAD(J,L,J)
      SADU(J,K,J)=SADU(L,J)
      SADU(J,K,J)=SADU(L,J)
      SACHU(J,K,J)=SACHU(L,J)
      SACHU(J,K,J)=SACHU(L,J)
      903 GO TO 107
      C ZERO OUT ALL OR PART OF RUFFERS
      C 900 00 901 L=KX,MXNPR
      NADJEL(L)=0
      NADJEL(L)=0
      ITYPE(L)=0
      THETAIL(0,0,0)
      R(L)=0,0
      Z(L)=0,0
      XMASS(L)=0,0
      SNUU(L)=0,0
      SNPW(L)=0,0
      SNPW(L)=0,0
      00 901 J=1,MAXJUP
      NPAD(JL,J)=0
      SADU(JL,J)=0,0
      SADU(JL,J)=0,0
      SACHU(JL,J)=0,0
      SACHU(JL,J)=0,0
      901 GO TO (2,108), ISWTC
      C CALCULATE MINIMUM DIAGONAL PERIOD OF OSCILLATION
      C 300 PI=3.1415927
      IF(DUMU.EQ.0.0) GO TO 330
      DUMU=2.*P/DSQRT(DUMU)
      330 IF(DUMU.EQ.0.0) GO TO 331
      DUMU=2.*P/DSQRT(DUMU)
      331 CONTINUE
      IU=NPTN(IW)
      IW=NPTN(IW)
      WRITE(6,301) IU,DUMU,IW,DUMU
      301 FORMAT(1H1,4SH SMALLEST MAIN DIAGONAL FREQUENCY (SF/C/CYC/L)/,
      112H NODE POINT= 15.5X.8M PERIOD= 1PE15.5/
      212H NODE POINT= 15.5X.8M PERIOD= 1PF15.5/
      PERIOD=DUMU(DUMU, DUMU)
      IF(DUMU.EQ.0.0) PERIOD=DUMU
      IF(DUMU.EQ.0.0) PERIOD=DUMU
      C REWIND 1
      REWIND 4
      REWIND 6
      REWIND 12
      RETURN
      END

      SUBROUTINE ELAST(IELAST,ISTRES,E1,E2,E3,E4,E5,C,NUME)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(1..4)

      C FORM STRESS-STRAIN MATRIX, F IN C
      C
      C 00 1 I=1,4
      C 00 1 J=1,4
      C C11,J=0,0
      C IF(IELAST.NE.1) GO TO 20
      C
      C (F(IESTRES,EQ.2)) GO TO 4
      C
      C ISOTROPIC AXISYMMETRIC OR PLANE STRAIN PROBLEM
      C
      EBAR=E1/((1.+E2)*(1.-2.*E2))
      C(1,1)=EBAR*(1.-E2)
      C(1,2)=EBAR*E2
      C(1,3)=C(1,2)
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,2)
      C(3,1)=C(1,2)
      C(3,2)=C(1,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
      C
      C ISOTROPIC PLANE STRESS PROBLEM
      C
      C EBAR=E1/(1.-E2*E2)
      C(1,1)=EBAR
      C(3,1)=EBAR*E2
      C(1,3)=C(3,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
      C
      C 20 IF(IELAST.NE.2) GO TO 30
      C
      C (F(IESTRES,EQ.2)) GO TO 2
      C
      C ANISOTROPIC AXISYMMETRIC OR PLANE STRAIN PROBLEM
      C
      C(1,1)=E1
      C(1,2)=E1-2.*E5
      C(1,3)=E3
      C(2,1)=C(1,2)
      
```

```

C(2,2)=C(1,1)
C(2,3)=C(1,3)
C(3,1)=C(1,3)
C(3,2)=C(2,3)
C(3,3)=E2
C(4,4)=E4
RETURN

C   ANISOTROPIC PLANE STRESS PROBLFM
C
C   2  C(1,1)=2.*E5*(E1-2.*E5)/E1
C     C(1,3)=2.*E3*E5/E1
C     C(3,1)=C(1,3)
C     C(3,3)=E2-E3**2/E1
C     C(4,4)=E4
RETURN

C   21 WRITE(6,3) IELAST,NUME,ISTRES
3  FORMAT(1H1/31H ERROR IN ELASTIC CONSTANT DATA/
1 13H IELAST      =,15H ELEMENT NO.,,15/
2 13H ISTRES     =,15)
CALL EXIT

C   COMPRESSIBLE FLUID
C
C   30 IF(IELAST.NE.3) GO TO 21
C   IF(ISTRES.EQ.2) GO TO 21
C
DC 31 I=1,3
  00 31 J=1,3
  31 C(I,J)=E1
RETURN
ENC

```

```

SUBROUTINE STIFF(KASE,NPI,NPJ,NPK,NPL,NUME,MAXNP,1STRES,C,R,Z,
1CK,AI,S1,C1)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION C(4,4),R(MAXNP),Z(MAXNP),CK(8,8),AI(23),D(8,8),G(8,8),
1VEC(8)

C   COMPUTE ELEMENT STIFFNESS MATRIX
C
C   KASE   = 1 GENERAL TRIANGLE
C         = 2 NODE 1 ON Z-AXIS
C         = 3 NODES 1, K ON Z-AXIS
C         = 4 GENERAL RECTANGLE
C         = 5 NODE 1 ON Z-AXIS
C         = 6 NODES 1, L ON Z-AXIS
C         = ELASTIC MODULI MATRIX
C   CK    = STIFFNESS MATRIX
C   AI    = INTEGRALS FOR COMPUTING K AND M
DO 1 I=1,8

```

```

A=DSQRT(AJ*AJ+RJ*RJ)
AL=R*(NPL)-R(NP1)
BL=Z*(NPL)-Z(NP1)
H=DSQRT(AL*AL+RL*RL)
H=A*B
IF(KASE.EQ.4) GO TO 6
C(1,1)=1.D
0(2,1)=-B/H
D(3,1)=1./H
D(4,1)=-A/H
6 D(5,2)=1.0
D(6,2)=-B/H
D(7,2)=1./H
D(8,2)=-A/H
0(2,3)=-C(6,2)
0(3,3)=-C(7,2)
0(6,4)=C(2,3)
0(7,4)=D(3,3)
0(3,5)=0(7,2)
0(7,6)=0(7,2)
IF(KASE.EQ.6) GO TO 7
0(3,7)=D(3,3)
0(4,7)=-0(8,2)
7 D(7,8)=D(7,4)
D(8,8)=-D(8,2)
NORD=8
C
IF(KASE.EQ.4) GO TO 8
IF(STRES.NE.0) GO TO 9
G(1,1)=C(2,2)*AI(5)
G(2,1)=C(1,2)*C1*AI(1)*C(2,2)*AI(7)
G(3,1)=C(1,2)*C1*AI(1)*(2+SI*AI(3))+C(2,2)*AI(6)
G(4,1)=C(1,2)*SI*AI(1)*(1+CI*AI(2))
G(6,1)=-C(2,3)*(CI*AI(3)-SI*AI(1))
G(7,1)=C(2,3)*(CI*AI(3)-SI*AI(2))
G(8,1)=C(2,3)*(CI*AI(1))
9 C
 8 DUM1=C1*AI(14)+SI*AI(13)
  DUM2=C1*AI(13)-SI*AI(14)
  DUM3=C1*CI*AI(12)+2.*SI*CI*AI((15)+SI*SI*AI(11))
  DUM4=SI*SI*AI(12)-2.*SI*CI*AI((15)+CI*CI*AI(11))
  G(2,2)=CI*(C1,1)*C1*AI(4)*2.*C1(2,2)*C1(2,2)*AI(1C)
  1+C(4,4)*SI*SI*AI(4)
  G(3,2)=C1,1)*C1*DUM1+C1,2)*(2.*C1*AI(16)+SI*AI(15))
  1+C(2,2)*AI(17)-C(4,4)*SI*DUM2
  G(7,2)=C1,3)*C1*DUM2+C1(2,3)*(C1*AI(16)-SI*AI(16))
  1-C(4,4)*SI*DUM1
  G(8,2)=AI(4)*(C1,3)*C1*CI-C1-C1*AI(4)*SI*SI*AI(3)
  G(3,3)=C1,1)*SI*SI*2.*C1(1,2)*(C1*AI(20)+SI*AI(23))
  1+C(2,2)*AI(21)+C1,4)*SI*DUM4
  G(6,3)=-C1,3)*SI*DUM1-C1,2)*SI*AI(16)+C(4,4)*C1*DUM2
  G(7,3)=(C1,3)+C(4,4)*(SI*SI*AI(11)-AI(11)-AI(12)+AI(15)*(C1*CI-SI*SI))
  1+C(2,3)*C1*AI(23)-SI*AI(20)
  G(8,3)=C1,3)*C1*DUM1+C1(2,3)*C1*AI(16)+C(4,4)*SI*DUM2
  G(6,6)=(C1,3)*SI*SI*C1(4,4)*C1*CI*AI(4)
  G(7,6)=-C(2,3)*SI*DUM2+C1(4,4)*C1*DUM1

```

```

G(7,7)=C(3,3)*DUM4+C(4,4)*DUM3
G(8,7)=C(3,3)*C1*DUM2+C(4,4)*C1*DUM1
G(8,8)=(C1,3)*C1*C1*(4,4)*SI*SI*AI(4)
H=A*B
IF(KASE.EQ.6) GO TO 3C1
G(4,2)=(C1,1)-C(4,4)*SI*CI*AI(4)*C(1,2)*(C1*AI(2)+SI*AI(3))
1+C(2,2)*AI(9)
G(6,2)=-(C1,3)+C(4,4)*SI*CI*AI((4)-C(2,3)*SI*AI(3)
G(4,3)=C1,1)*SI*DUM1+C1(1,2)*(C1*AI(19)+2.*SI*AI(16))
1+C(2,2)*AI(22)+C1*CI*AI(2)*C(2,2)*SI*AI(2)
1+C(4,4)-C1,1)*SI*SI*AI(4)*DUM2
1+C(4,4)*C1*CI*AI(4)
G(6,4)=AI(4)*(C1(4,4)*C1*CI-C1,3)*SI*SI)-C(2,3)*SI*AI(2)
G(7,4)=C1,3)*SI*DUM2+C1(2,3)*C1*AI(16)-SI*AI(19)+r(4,4)*C1*DUM1
G(8,4)=(C1,3)*C1*(4,4)*SI*CI*AI(4)*C(2,3)*C1*AI(2)
G(8,6)=SI*CI*(C1(4,4)-C1(3,3)*AI(4)
C
C FILL IN REST OF G AND CALCULATE K
C
301 00 201 I=2,NORD
K=1-1
00 201 J=1,K
201 G(J,I)=C(I,J)
C
00 51 J=1,NORD
00 50 L=1,NORD
VECIL)=0.0
00 50 K=1,NORD
50 VECIL)=VECIL+G(L,K)*DI(K,J)
DO 51 I=1,NORD
CK(I,J)=0.0
DO 51 L=1,NORD
DO 51 K=1,NORD
51 CK(I,J)=CK(I,J)+DI(L,J)*DI(L,I)*VECIL(L)
C
RETURN
END
C
SUBROUTINE INS(KASE,NP,NPJ,NPK,NPL,ISTRES,R,Z,AL,SI,CI,MAXNP)
IMPLICIT REAL*8(A-H,O-Z)
DIMENS(IN N AI(123),R(MAXNP),Z(MAXNP))
C COMPUTE ELEMENT INTEGRALS
C
00 1 I=1,23
1 AI(1)=0.
C
IF(NPL.NE.0) GO TO 3RC
C
TRIANGULAR ELEMENT
C
AJ=R(NPJ)-R(NP1)
AK=R(NPK)-R(NP1)
BJ=Z(NPJ)-Z(NP1)
BK=Z(NPK)-Z(NP1)
H=AJ*PK-AK*PJ
R=BJ-PK

```

```

A=AJ-AK
R1=RINP1
C
IF(ISTRES.EQ.0.01 GO TO 100
A(4)=H/2.
A(13)=H*(AJ+AK)/6.
A(14)=H*(BJ+BK)/6.
RETURN
C
2 A(1)=H/2.
A(2)=H*(BJ+BK)/6.
A(3)=H*(AJ+AK)/6.
A(4)=R1+A(1)+A(3)
IF(KASE.EQ.11 GO TO 3
A(9)=A(2)
A(10)=A(4)
C
3 1COUNT=1
RA=R1
R=RINP1
C=B/J/AJ
0=0.
0UM=-1.
IF(C.EQ.0. ( GO TO 100
101 IF(KASE.EQ.11 GO TO 102
IF(KASE.NE.21 GO TO 104
IFRA.NE.0. ( GO TO 102
FC=LOG(RA)
GO TO 104
102 FC=LOG(RA/RA)
DUM1=R-B-A
DUM2=R*B-R*A
DUM3=R*B*D-R*A*R
DUM4=DIM2*(R*B-R*A)
IF(KASE.EQ.31 GO TO 103
F1=DUM1-R*FO
F2=DUM2*-2.*R*(DUM1*R*FO
F3=DUM3/3.-1.5*R1*DUM2*3.*R*(R*(DUM1-R*(R1*FO
103 GO=DUM2/2.
G1=DUM3/3.-R1*DUM2/2.
G2=DUM4/4.-2.*R1*DUM2/3.*R*(R1*DUM2/2.
C
IF(KASE.NE.1) GO TO 105
A(5)=A(15)+DUM1*(C*F1*D+F1*(C*F2/2.*C*D*F1*D+F1*D/2.)
A(6)=A(16)+DUM1*(C*C*F3/3.+C*C*D*F2+C*D*F1*D+F1*D/2.)
A(7)=A(17)+DUM1*(C*F2*D+F1)
A(8)=A(18)+DUM1*(C*c*F3/2.*C*D*F2*D+F1*D/2.)
A(10)=A(10)+DUM1*(C*D*F3*D+F2)
GC TO 1.6
105 IF(KASE.NE.2) GO TO 107
106 A(1)=A(8)+DUM1*(C*C*F3/3.+C*C*D*F2+C*D*F1*D+F1*D/2.)
107 A(13)=A(13)+DUM1*C*G2*D*G1
A(14)=A(14)+(DUM1*(C*C*G2/2.*C*D*G1*D+F1*D/2.)
C
108 GO TO 201,202,203,1 COUNT
201 (COUNT=2

```



```

C
C      RAK1INC,IND+1)=0.0
C      RAK1INC+1,IND+1)=SFAC
C
C      20 DC 100 J=1,LIM
C      IF(1.EQ.J) GO TO 100
C      JIN=MNODE(J)
C      JMC=2*J-1
C
C      IF(.NOT.(ITYPE(EINI)).EQ.2.0R.+ITYPE(F(JNI)).EQ.211 GO TO 4C
C
C      RAK1INC,JND)=0.0
C      BK1(IND+1,JND)=0.0
C      BK1INC,JND+1)=0.0
C      BK1(IND+1,JND+1)=0.0
C      GO TO 100
C
C      50 IF(ITYPE(EINI).NE.11 GO TO 75
C
C      S=DSIN(THETAIJN1)
C      C=DCOS(THETAIJN1)
C      BK1(IND,JND)=BK1(IND,JNN)*C+RK1(IND+1,JND)*S
C      BK1(IND,JND+1)=RK1(IND,JNN+1)*C+RK1(IND+1,JND+1)*S
C      BK1(IND+1,JND)=0.0
C      BK1(IND+1,JND+1)=0.0
C
C      75 IF(ITYPE(EINI).NE.11 GO TO 100
C
C      S=DSIN(THETAIJN1)
C      C=DCOS(THETAIJN1)
C      BK1(IND,JND)=BK1(IND,JNN)*C+RK1(IND+1,JND)*S
C      BK1(IND+1,JND)=BK1(IND,JNN+1)*C+RK1(IND+1,JND+1)*S
C      BK1(IND+1,JND+1)=0.0
C
C      100 CONTINUE
C
C      RETURN
C
C
C      AJ=R(NPJ)-Q(NP1)
C      AK=P(NPK)-P(NP1)
C      RJ=Z(NPJ)-Z(NP1)
C      RK=Z(NPK)-Z(NP1)
C      H=AJ*OPU-AK*OJP
C      B=BJ-PK
C      A=AJ-AK
C
C      DUM1=RNG/H
C      AM1=DUM1*(A-EAI(14))-AK*EAI(14)
C      AM2=DUM1*(A-EAI(14))-RJ*EAI(14)
C      GO TO 3
C
C      2 AJ=R(NPJ)-R(NP1)
C      RJ=Z(NPJ)-Z(NP1)
C      A=DSORT(AJEAJ*BL*ORJ)
C      AL=AMPL-R(NP1)
C      BL=Z(NPL)-Z(NP1)
C      B=DSORT(AL*AL+BL*BL)
C
C      TPI1STRES=.50.01 GO TO 4
C
C      AM1=RHO*EAI(4)/4.
C      AMJ=AP1
C      AMK=AM1
C      AML=AM1
C      GO TO 3
C
C      4 DUM1=RHO/(1.+.51+C1)
C      AM1=DUM1*(S1+C1)*EAI(4)-S1*EAI(14)/M-C1*EAI(13)/A1
C      AMJ=DUM1*(T1+C1)*EAI(4)-(1.+C1)*EAI(14)/M-C1*EAI(13)/A1
C      AMK=DUM1*(T1+C1)*EAI(4)+(1.+C1)*EAI(14)/B+(1.+S1)*EAI(13)/A1
C      AML=DUM1*(T1+C1)*EAI(4)+S1*EAI(14)/A-(1.+S1)*EAI(13)/A1
C
C      3 XMASS(NPL)*XMASS(NP1)*AM1
C      XMASS(NPJ)*XMASS(NP1)*AMJ
C      XMASS(NPK)*XMASS(NP1)*AMK
C      IF(NPL.EQ.0) RETURN
C      XMASS(NPL)*XMASS(NP1)*AML
C
C      RETURN
C
C
C      SUBROUTINE PONK(MAXNP,MADJJP,MADJNP,MADJL,IPRINT,SNPUU,
C      ISNPUM,SNPNU,SADNU,SADUM,SADWW,THEIA,ITYPE,XMASS,MPUT,
C      2NUMP1)
C      IMPLICIT REAL(8)-H,O-2)
C      DIMENSION AI(40)
C      IXMASS(MAXNP),
C      2SAQUIMAXP,MXANJP1,SADNU,MADJJP,SNPNU,MAXNP,MAXNP,
C      3SAQCHIMAXP,MXANJP1,THTAIMAXP1,ITYPE(MAXNP)
C
C      PAINT STIFFNESS AND MASSES
C
C      DATA AI/6MBEFORE,6MAFTER,6MSTIFFN,6MESS,
C      16MBTRIND,6MRY ALT,6HEPATD,6AN
C      6M MNF,6MSNPUI,6MSADNU,6MSADWW,6MSSTANU,6MSSTRANU,6MSSTARRYU,6MSSTARRYP,
C      36MSTRESS,6MSTTPU,6MSTSPU,6MSTTPW,6MSTSPW,6MSTTARYU,6MSTTARYP,6MSSTARRYU,6MSSTARRYP,
C      46M TSNU,6MSTSNP,6MSTSPU,6MSTTPU,6MSTTPW,6MSTSPW,6MSTTARYU,6MSTTARYP,6MSSTARRYU,6MSSTARRYP
C      56MSTZADW,6MSTSDW,6MSTSDW,6MSTCP,6MHP-SX/1./6M

```



```

6 NPI=NPI-NPOUT
NPK=NPK-NPOUT
NPK=NPK-NPOUT
IFINIT=EQ.01 NPL=0
IFINIT=EQ.01 NPL=0
CALL STRESSMP,STADU,C,NPI,NPJ,NPK,NPL=0,Z,MASF,MUF,MUW,
| STMPU,STMPU,STADU,STADU,ISRES,SI,CI,NPOUT,ITYPE,THETA
| IF(ICOUNT,L1,MUNFL) GO TO 4
KEND=1
KEY=MADJJP+1
C 100 MUPCPKEY-1
MUPCP=MUPCP-MPOUT
113 CALL MODS1JXMPB,MKAJP,MADJJP,MPAOJL,STMPU,STMPU,
| STADU,STADU,ITPAINT,THETA,ITYPE,NPOUT,MUMMP!
DO 101 I=1,MUMMP
101 WRITE(31,I,MADJNP(I)),MADJEL(I),IMPADJ(I,J),J=1,MADJJP
| ,STMPWIK,I,K=1,4),((STANWIK,J,J),STANWIK,J,J=1,MADJJP)
C IF(KEND,EQ. 1) GO TO 370
C
NPA=MUPCP-MUMMP
GO TO 902
C
107 KK=NPA+1
ISWTC=2
GO TO 900
C
108 IF((MUPCP-MUMMP).LT.MUPCP) KNP=MUMMP-MUPCP-NPA
| IF(MUPCP-MUMMP).GE.MUPCP KNP=MUMMP-KNP
| DO 109 I=1,KNP
L=NPA+1
READ(I,NPA,MADJNP(I)),MADJFL(I),(MPAOJL,J),J=1,MADJJP)
READ(I,NPA,RCL,I,Z,I,TYPEFL,I,THTFL,I,THTFL)
109 CONTINUE
NPOUT=MUMCP
GO TO 6
C
902 DO 903 K=1,NPA
L=MADJNP(K)
MADJNP(K)=MADJNP(L)
MADJEL(K)=MADJEL(L)
TYPEFL(K)=TYPEFL(L)
THEFL(K)=THEFL(L)
RFL=RFL
ZFL=ZFL
DO 904 J=1,4
STMPU(J,K)=STMPU(J,L)
904 STMPU(J,K)=STMPU(J,L)
DO 903 J=1,MKAJP
NPADJ(K,J)=MPAOJL,J
903 STADU,X,J,J=STADU,I,L,I
903 STADU,X,K,J,J=STADU,I,L,I
GO TO 107

```

```

1 IF (NPL,NE.0) GO TO 200
C   TRIANGULAR ELEMENT
C
C   AJ=RA(NPJ)-RA(NP1)
C   AK=RA(NPK)-RA(NP1)
C   BJ=Z(A(NPJ)-Z(A(NP1)
C   BK=Z(A(NPK))-Z(A(NP1)
C   H=A(J-BK-AK*BK)
C   B=BJ-BK
C   A=AJ-BK
C
C   25 IF (KASE,NE.1) GO TO 26
C   IF (LSTRES,NE.0) GO TO 10 27
C   ANG=1./RR
C   ROR/RR
C   ZOR=L/RR
C   GO TO 26
C
C   27 IF (ICOUNT,NE.1) GO TO 100
C   ANG=0.0
C   ROR=0.0
C   ZOR=0.0
C
C   28 DUM1=B/H
C   DUM2=A(H*ROR-A*ZOR)/H
C   OC 35 I=1,3
C   35 S(I,1)=C(1,1)*DUM1*C(1,2)*DUM2
C   S(I,4,1)=C(4,4)*A/H
C   GO TO 29
C
C   26 ROR=1.0
C   IF (KASF,NE.2) GO TM 3C
C   IF (ICOUNT,NE.1) GO TO 11
C   FOR=0.0
C   GO TO 29
C   31 ZOR=L/RR
C   GO TO 29
C   30 FOR=0.0
C   IF (ICOUNT,NE.1) GO TM 100
C   29 00 36 I=1,3
C   36 S(I,1,2)=C(1,1,3)*A/H
C   S(I,4,2)=C(4,4)*B/H
C   DUM1=BK/H
C   DUM2=(BK*ROR-A*K*ZOR)/H
C   00 37 I=1,3
C   37 S(I,1,3)=C(1,1,1)*DUM1*C(1,2)*DUM2
C   DUM1=AK/H
C   S(I,4,3)=C(4,4)*C(1,1,3)*DUM1
C   DO 38 I=1,3
C   38 S(I,1,4)=C(1,1,3)*DUM1
C   S(I,4,4)=C(4,4)*BK/H
C   DUM1=AJ/H
C
C   0C 39 I=1,3
C   39 S(I,1,6)=C(1,1,3)*DUM1
C   S(I,4,6)=C(4,4)*C(1,1,4)*DUM1
C   KASF=EQ.31 GO TM 30;
C   DUM1=-PJ/H
C   DUM2=(AJ*ZOR-BK*ROR)/H

```

A-19

35

```

69 S(1,5)=C1,1)*DUM1+C(1,2)*DUM2
  DUM1=(R*P*Cl-Z*P*S1)/H
  S(4,5)=C14,4)*DUM1
  DO 70 I=1,3
70 S(1,6)=C11,3)*DUM1
  S(4,6)=C14,4)*(Z*P*Cl+R*P*S1)/H
  DUM1=(A*Cl-R*P*Cl+Z*P*S1)/H
  00 71 I=1,3
71 S(1,8)=C11,3)*DUM1
  S(4,8)=C14,4)*(A*Cl-R*P*S1-Z*P*Cl)/H
  IF(KASE.EQ.6) GO TO 300
  S(4,7)=C14,4)*DUM1
  DUM1=(A*Cl-R*P*S1-Z*P*Cl)/H
  DUM2=(A*Z*OR-R*OR*Z*P1)/H
  00 72 I=1,3
72 S(1,7)=C11,1)*DUM1+C(1,2)*DUM2
C  MDO(FY FOR ROLLER SUPPORT
C  300 IF(I*TYPE(INP1).NE.1) GO TO 301
  NP=NPI
  KCOUNT=1
  MX=1
  GO TO 350
301 IF(I*TYPE(INP1).NE.1) GO TO 302
  NP=NPI
  KCOUNT=2
  MX=3
  GO TO 350
302 IF(I*TYPE(INPK).NE.1) GO TO 303
  NP=NPK
  KCOUNT=3
  MX=5
  GO TO 350
303 IF(INPL.EQ.0) GO TO 100
  IF(I*TYPE(INPL).NE.1) GO TO 100
  NP=NPL
  KCOUNT=4
  MX=7
  C 350 NX=NX+1
  CSN=DCOS(THETA(INP))
  SSP=DSIN(THETA(INP))
  DO 351 J=1,4
  SJ,J,MX)=SI(J,MX)*CSN+SS(J,NX)*SSN
  351 SI(J,NX)=0.0
  C  GO TO (301,303,100),KCOUNT
  C  DIST(PUTE ELEMENT STRESS IN ONE POINT STRESS
  C  100 IF(I*COUNT.NE.1) GO TO 101
  K=1
  KJ=3
  NK=5
  NL=7
  GO TO 103
C  101 IF(I*COUNT.NF.=2) GO TO 102
  K=3
  NJ=1
  NK=5
  NL=7
  N=NP
  NJ=NP1
  NK=NPK
  NL=NPL
  GO TO 103
C  102 IF(I*COUNT.NE.3) GO TO 107
  K=5
  NJ=1
  NK=3
  NL=7
  NI=NPK
  NJ=NPI
  NK=NPK
  NL=NPL
  GO TO 103
C  107 NI=7
  MJ=1
  NK=3
  NL=7
  NI=NPK
  NJ=NPI
  NK=NPK
  NL=NPL
  GO TO 103
C  103 DO 114 I=1,4
  STNPUI(I,NI)=STNPUI(I,NI)+SI(I,M)
  114 STNPWI(I,NI)=STNPWI(I,NI)+SI(I,M(+1))
  N=1
  115 GO TO (1150,140,120,153),N
  116 STNPWI(I,NI)=STNPWI(I,NI)+SI(I,M(+1))
  NN=NJ
  MN=MJ
  GO TO 154
  151 NN=NK
  MN=MK
  GO TO 154
  152 IF(INPL.EQ.0) GO TO 153
  NN=NL
  MA=ML
  C  154 I= 104 K=1,MYANJP
  J=K
  IF(INPADJ(N,K)-NPUT1),FO,NN) GO TO 155
  155 COUNT(NF)

```



```

C   ICOUNT=ICOUNT+1
C   IF(ICOUNT.LE.NUMCLS) GO TO 101
C
C   WRITE NPTN ONTO TAPE 10
C
C   WRITE ITF(10) INPTN(I), I=1,NUMNP)
C
C   READING STRESS TABLES FROM TAPE 3 AND WRITE BLOCKS ONTO TAPE 10
C
C   ICOUNT=1
201  NCLOW=NPLOW(ICOUNT)
      NHGH=NPHIGH(ICOUNT)
      DO 200 L=NLIN, NHGH
      I=L-NLOW+1
200  READ(3) NPM,NAOJNP(I),NAOJEL(I),(NPADJ(I,J)*J=1,MXADJP),
      ISTNPUIK(I,STNPWIK,I),K=1,4),(ISTADUIK,I,J)*STANWIK,I,J,
      2K=1,4),J=1,MKADJP)
C
      NUMNP=NHGH-NLOW+1
      WRITE(10) NLIN,NHGH,NUMNP,
      1NAOJNP(I),NAOJEL(I),(NPANJ(I,J)*J=1,MXADJP),
      1STNPUIK(I,K=1,4),(ISTADUIK,I,J)*STANWIK,I,J,*K=1,4),J=1,MXADJP).
      3I=1,NUMNP)
C
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.NUMCLS) GO TO 201
C
C   203  CONTINUE
      END FILE 10
      REWIND 10
      REWIND 3
      RETURN
C
C   CALCULATE MINIMUM DIAGONAL PERIOD OF OSCILLATION FOR RFRUN
C
C   300  DUMMY=0.0
      REWIND 1C
      REA010)
305  READ(10) N1,N2,
      N5,(NAJDNP(I),ITYPE(I)),THETAI),XMASSII),
      1SNPUII)*SNPUW(I),SNPWII),(NPADJ (I,J)*SAOUU(I,J)*SADUW(I,J),
      XSAOUW(I,J),
      2SAOUW(I,J)*J=1,MKA0JP),I=1,N5)
      00 301 I=1,N5
      DUMMY1=0.0
      IF(ITYPE(I)=NE_2_ANO_ITYPE(I)).NF.=1) DUMMY2=SNPNW(I)/XMASSII)
      DUMMY=OMAX1(DUMMY,DUMMY1,DUMMY2)
C
      301  CONTINUE
      IF(N2.LT.NUMNP) GO TO 305
      REWIND 10
      P1=3,1415927
      PERIOD=2.*P /OSQRT(DUMMY)
      WRITE(16,303) PERIOD
      303 FORMAT(1H1,1MPER(0D=,1PF15.5,X,9HSEC/VCLF)

```

```

NUMCEL=0
NCLUS=0
C 100 IF((NUMPEL-NUMCEL).LT.MAXEL) NUMFLB=NUMPEL-NUMCEL
      (FI((NUMPEL-NUMCEL).GE.MAXEL) NUMELR=MAXELR
C DO 200 KK=1,NUMELR
C
C READ(14) NODEF(KK),IZONE,(PLAST(KK),NP(KK,1),NP(KK,2),NP(KK,3),
      INP(KK,4),ITYPE(KK,1),ITYPE(KK,2),ITYPF(KK,3),ITYPF(KK,4),
      2 THETAKK,1),THETA(KK,2),THETA(KK,3),THETA(KK,4),
      3 R(RJRK,RL,ZI,ZJ,ZK,ZL
C
C II=1
201 NODE=NP(KK,1)
      NP(KK,1)=NPTP(NODE)
      II=II+1
      IF(II.LT.4) GO TO 201
      IF(II.GT.4) GO TO 202
      IF(NP(KK,4).EQ.0) GOTO 202
      GC TO 201
C
C 202 I=E LAST(IZONF)
      A1=E1(IZONE)
      A2=E2(IZONE)
      A3=E3(IZONE)
      A4=E4(IZONE)
      A5=E5(IZONE)
      NUME=NODEF(KK)
      CALL ELOST(1E,1STRES,A1,A2,A3,A4,A5,CC,NUME)
      DO 203 I=1,4
      DO 203 J=1,4
      C(KK,I,J)=CC(I,J)
C
C 204 DO 204 I=1,4
      DO 204 J=1,8
      PIKK,J,J)=0.0
      204
C IF(INP(KK,4).NE.0) GO TO 205
C
C AJ=RJ-R1
      AK=RK-R1
      BJ=ZJ-Z1
      BK=ZK-Z1
      HJ=AJ*RK-AK*RJ
      AA=AJ-AK
      BB=BJ-BK
C
C PIKK,1,1)=RB/HH
      PIKK,1,3)=RK/HH
      PIKK,1,5)=BJ/HH
      (F1STRES.NE.0) GO TO 205
      RC=(AJ+AK)/3.
      ZD=(BJ+BK)/3.
      CAPRO=R1+R0
      205
      BIKK,2,1)=(HH+RA*RD-AA*ZD)/(HH*CAPRO)
      BIKK,2,3)=(BK*RC-AK*ZD)/(HH*CAPRO)
      BIKK,2,5)=(-BJ*RN+AJ*ZD)/(HH*CAPRO)
      205
      BIKK,3,2)=-AJ/HH
      BIKK,3,4)=-AK/HH
      BIKK,3,6)=AJ/HH
      BIKK,4,1)=BK*KK,3,2)
      BIKK,4,2)=BK*KK,1,1)
      BIKK,4,3)=BK*KK,3,4)
      BIKK,4,4)=RK*KK,1,3)
      BIKK,4,5)=RK*KK,3,6)
      BIKK,4,6)=RK*KK,1,5)
C
C IF((1STRES.NE.0) CONST=HH/2.
      IF((1STRES.FQ.O) CONST=HH*CAPRO/2.
C
C 212 DC 206 (=1,8
      DO 206 J=1,4
      DO 207 N=1,4
      207 PIKK,I,J)=PIKK,I,J)+CONST*B(KK,N,(1)*CC(N,J)
      206
      CONTINUE
      GO TO 231
C
C
C 208 AJ=RJ-R1
      BJ=ZJ-Z1
      AA=DSQRT(AJ*AJ+RJ*RJ)
      AL=RL-RI
      RL=ZL-Z1
      RB=DSQRT(IAL*AL+BL*BL)
      MH=AA*RB
      SJ=-BJ/AA
      CJ=AJ/AA
C
C
C 209
      IF((1STRES.FQ.O) GO TO 209
      RC=AA/2.
      2C=BB/2.
      GO TO 210
      209
      AI NT1=HH
      AI NT2=HH*BR/2.
      AI NT3=HH*AA/2.
      AI NT4=HH*(RI+(AA*CL+BR*SL)/2.)
      AI NT13=AA*AI NT4/2.+HH*AA**2*CL/12.
      AI NT16=(I-H/2.)**2*CL/12.
      AI NT16=(I-H/2.)**2*CL/12.
      RC=AI NT13/A(IN T4
      ZD=AI NT14/A(IN T4
      DUMMY=RO*SL+ZO*CL
      BIKK,1,1)=(-AA*SI-BR*CI+DUMMY1)/HH
      BIKK,1,3)=(RBCI-DUMMY)/HH
      BIKK,1,5)=DUMMY/HH
      BIKK,1,7)=(AA*SI-DUMMY)/HH
      IF((1STRES.NE.0) GO TO 211
      RIKK,2,1)=(HH*AI NT2+A(IN T3+A(IN T2)/(HH*AI NT4),
      BIKK,2,3)=(BB*AI NT2-A(IN T16)/(HH*AI NT4),
      BIKK,2,5)=A(IN T16/(HH*AI NT4)
      210
      40

```

```

B(KK,2,7)=(AA*AINT2-A(INT16)/(INT16)*AINT4) C
C
211 DUMMY=R0*CL-Z0*S1
A(KK,3,2)=(R0*S1-AA*C1)+DUMMY)/HH
B(KK,3,4)=(-R0*S1-DUMMY)/HH
B(KK,3,6)=DUMMY/HH
B(KK,3,B)=(AA*C1-DUMMY)/HH
B(KK,4,1)=B(KK,3,2)
B(KK,4,2)=B(KK,1,1)
B(KK,4,3)=B(KK,3,4)
B(KK,4,4)=B(KK,1,3)
B(KK,4,5)=B(KK,3,6)
B(KK,4,6)=B(KK,1,5)
B(KK,4,7)=B(KK,3,B)
B(KK,4,B)=B(KK,1,7)
IF(ISTRS.NE.0) CONST=HH
IF(ISTRS.EQ.0) CONST=AINT4
GO TO 212
C
C
231 DO 232 I=1,4
EPST11(KK,I)=0.0
EPSPI11(KK,I)=0.0
232 SIG11(KK,I)=0.0
C
IF(IPLAST(KK).NE.1) GO TO 237
C
I1UM(KK,21)=NOY1LO(IZONE)
KY1L0=NOY1LD(IZONE)
00 233 I=1,KY1L0
DUMIKK,I)=SSSTAR(IZONE,1)
233 DUMIKK,I+10)=HSTAR(IZONE,1)
DO 234 I=1,B
234 DUMIKK,I+21)=0.0
DUMIKK,27)=SSSTAR(IZONE,1)
GO TO 200
C
237 IF(IPLAST(KK).NE.2) GO TO 235
C
DUMIKK,1)=ALPHA(IZONE)
DUMIKK,2)=CAPPA(IZONE)
DUMIKK,3)=COSTH(IZONE)
1DUMIKK,4)=0
IF(CAPPA(IZONE).EQ.0.0) I1UMIKK,4)=1
00 238 DUMIKK,I+3)=0.0
238 DUMIKK,I+4)=0.0
GO TO 200
C
C
235 WRITE(6,236) NOOFFL(K),IZONE,IPLAST(IKK)
236 FORMAT(1H1,28HERROR IN ELEMENT DATA, LNK1F//,
110X,12HELEMENT NO.=,15/10X,12HZONE NO. =,15/
210X,12H(PLAST =,15)
CALL EXIT
C
C
200 CONTINUE
C
C
4 EBAR=E1/(1.-E2*E2)
C(1,1)=EBAR
C(3,1)=EBAR+E2
C(1,3)=C(3,1)
C(3,3)=C(1,1)
C(4,4)=EBAR*(1.-E2)/2.
RETURN
C
C
4 EBAR=E1/(1.-E2*E2)
C(1,1)=EBAR
C(3,1)=EBAR+E2
C(1,3)=C(3,1)
C(3,3)=C(1,1)
C(4,4)=EBAR*(1.-E2)/2.
RETURN
C
C
41 20 IF(IELAST.NE.2) GO TO 30

```

```

21 WRITE(6,31) IELAST,NUME,ISTRES
3 FORMAT(1H1/31H ERROR IN ELASTIC CONSTANT DATA/
113H IELAST      * ,15/13H ELEMENT NO.=,15/
213H ISTRES      * ,15)
CALL EXIT
C
30 IF(IELAST.NE.3) GO TO 21
C
IF(ISTRES.FQ.2) GO TO 21
C
00 31 I=1,3
00 31 J=1,3
31 C(I,J)=E1
C
RETURN
ENC

```

```

SUBROUTINE LINK2
IMPLICIT REAL*16(A-H,O-Z)
COMMON MAXNP,MXDP,MXLNE,MXLND,MXNPD,MAXELR,NUMNP,NUMEL,
1  ISTRES,LINES,NUMPEL,LOADNP11,PFRIO,DT,NLOAD11,50),
2  RAD1,50,JZAD1,50,SNORM11,50),TITFL10),MAXBD,
3  MBLOCK,MREADS,MAXBK,TFRAL,PDAMP,RETA,KRUN,
4  TPELTP,KOUNT,IFILLAT,TMAX,TRERUN
DIMENSION COM11981
EQUIVALENCE (MAXNP,COM11981)

```

```

THIS LINK INTEGRATES THE EQUATIONS OF MOTION
READ AND PRINT INTEGRATION DATA
READIS,1000,TMAX,TRERUN,ET,KDT,KINT,BETA,PDAMP
1000 FORMAT(3E10.0,215,2E10.0)

```

```

TMAX =MAX. RUN TIME FOR PROBLEM (SEC.)
TRERUN=BEGINNING TIME FOR RESTART 1 SEC.)
ET =TIME INCREMENT (SEC.)
KDT =>0 USES DT AS READ IN
=>1 USES KINT TO CHOOSE DT
KINT =>DIVISOR FOR CHOOSING DT
BETA =>INTEGRATION SCHEME PARAMETER
PDAMP =>PERCENT CRITICAL DAMPING FOR ARTIFICIAL VISCOSITY
IF(BETA.LE.0.0.RBETA.GT.0.25) RBETA=1.0.C/6.0
REWIND A
IF(KRUN.LT.0) GO TO 10
READ(BITLFF
READIS)CM
GO TO 20
10 WRITE(BITLFF
D1=ET
IF(KDT.NE.0) D1=PERIOD/FLDATIKINT)
NREADS=1
IF(NUMLP.NE.0) NREADS=2+INUMPFL-1)/MAXFLB
TREAL=0.0

```

```

WRITE(6)COM
20 WRITE(6,2000)TMAX,TRERUN,OT,BETA,PDAMP
2000 FORMAT(1H1,30HMAX. TIME DURATION
1          1X,30HINITIAL TIME FOR INTEGRATION
2          1X,30HTIME INCREMENT
3          1X,30HRETA
7          1X,30HPODAMP
8          1X,3HPCT)

```

```

C CHECK LIMITS
C
IF(IKRUN.EQ.0.ANO.TMAX.FQ.0.0) STOP
40 IF(MBLOCK.L.E.MAXMBK) GO TO 50
WRITE(6,3000)MBLOCK,MAXBK
3000 FORMAT(49H1BANDWIDTH TOO LARGE FOR INTEGRATION CALCULATION./
:14H RADIWIDTH IS ,15,31H, MAXIMUM BANDWIDTH ALLOWED IS ,15)
STOP

```

```

C CALCULATE MATRICES FOR RFCURSION
C
50 IF(IKRUN.GT.1) GO TO 60
CALL LINK2A
C
INTEGRATE PRBLFM
C
60 IF(TMAX.NE.0.0) GO TO 70
END FILE B
REWINC B
STOP
70 CONTINUE
CALL LINK2B
C
RETURN
END

```

```

SUBROUTINE MMATIA,L,M,N,B,IA,IB,C)
IMPLICIT RFAL(BIA-H,0-Z)
DIMENSION A(L,M),R(M,N),C(L,N)
C
MATRIX MULTIPLY WITH TRANSPOSE OPTION
C
DO 10 IL=1,L
DO 10 IM=1,N
DO 10 IM=1,M
IL=IL
IF(IL.LT.0) IL=IM
I2=IL+IM-1
I3=IM
IF(IL.BLT.0) I3=IM
I4=IM+IM-1
C11,IN=C(I,L,(IN)+A(I,(2)*R(I,14))
10 RETURN
FNC

```

```

SUBROUTINE ERASF(N,X)

```

```

      REAL*8 X(IN)
      C
      ERASE N WORDS IN X
      C
      DO 10 I=1,N
      10 X(I)=0.0
      RETURN
      ENC

```

```

      DMP(2,I)=DAWU
      F1(1,I,I)=10.5*OAMU*OT*XMASS(I,I)/(BETA*OT*DT)+SNP(I,I,1)
      F1(2,I,I)=SNP(I,I,2)
      F1(3,I,I)=SNP(I,I,2)
      F1(4,I,I)=(C+5*OAMW*DT*XMASS(I,I)/(BETA*OT*DT)+SNP(I,I,3))
      C
      CHECK ADJACENT STIFFNESS TABLES FOR NODES IN THIS BLOCK
      C
      DO 945 K=1,MXADJP
      JNP=NPACJ(I,K)
      IF(JNP.LT.NI.OR.JNP.GT.N2) GO TO 945
      J=JNP-MI+1
      C
      COPY ADJACENT STIFFNESS INTO F1 AND ZERO IT OUT IN TABLES
      C
      F1(1,I,J)=SA0(I,K,1)
      F1(2,I,J)=SA0(I,K,3)
      F1(3,I,J)=SA0(I,K,2)
      F1(4,I,J)=SA0(I,K,4)
      00 920 M=1,4
      SA0(I,K,M)=0.0
      NACJNP(I)=NADJNP(I)-1
      NPA0J(I,K)=0
      C
      ZERO OUT ITS TRANPOSE ACROSS THE MAIN DIAGONAL
      C
      00 925 K=1,MXADJP
      KK=K
      IF(INPADJ(J,KL).EQ.INP) GO TO 930
      C
      925 CONTINUE
      WRITE(6,1000)INP,JNP
      1025 FORMAT(13HMITABLE SEARCH ERROR IN LINK2A. INP=,15.6H, JNP=,15)
      STOP
      930 OC S35 M=1,4
      935 SA0(I,K,M)=0.0
      NACJNP(I)=NADJNP(I)-1
      NPA0J(J,KK)=0
      945 CONTINUE
      950 CONTINUE
      C
      READ(10)INI,N2,NS,(NADJNP(I),ITYPE(I),THFTA(I,I),XMASS(I,I),SNP(I,I))
      IJ=1,3,(INPA0J(I,J),(SA0(I,J,K),K=1,4),J=1,MXADJP),I=1,NS
      C
      READ ANOTHER BLOCK FROM STIFFNESS TAPE
      C
      100 READ(10)INI,N2,NS,(NADJNP(I),ITYPE(I),THFTA(I,I),XMASS(I,I),SNP(I,I))
      IJ=1,3,(INPA0J(I,J),(SA0(I,J,K),K=1,4),J=1,MXADJP),I=1,NS
      C
      ZERO OUT F1
      CALL ERASF(4*MAXMBK,F1)
      CCPY C(N) INTO F1
      DO 950 I=1,NS
      INP=I+NS-1
      C
      ACC DIAGONAL SFT OF EQUATIONS
      C
      OAMU=2.0*DSORT(XMASS(I,I)*SNP(I,I))*POAMP/100.0
      OMP(I,I)=DAWU
      DAWU=2.0*DSORT(XMASS(I,I)*SNP(I,I))*PNAMP/100.0
      L=NPACJ(I,K)

```

```

IF(F1.L.EQ.0.OR.L.GE.N1) GO TO 120
L=L-NILAST+1
DO 121 K2=1,MAXJP
  NPAC(J,K2)
  IF(N.EQ.0.OR.N.GE.N1) GO TO 121
  NILAST+1
C SUBTRACT OFF CONTRIBUTION
C
  IL=MINDIL,LI
  IMAXDIL,MI
  TMP11=-F1(L,IL,IMI)
  TMP12=-F1(2,IL,IMI)
  TMP131=-F13,IL,IMI
  IF(1M-LI.GF.,0) GO TO 125
  SAVE(TMP131)
  TMP131=TMP121
  TMP141=-F14,IL,IMI
  F111,L,JI=F111,L,J1+SA(D(J,K1,1)*SA(D(J,K2,1)*T(MP11)*
  1 SA01(L,K1,2)*SA01(J,K2,1)*T(MP12)*
  2 SA01(L,K1,1)*SA01(J,K2,2)*T(MP14)*
  3 SA01(L,K1,2)*SA01(J,K2,3)*T(MP11)*
  1 SA01(L,K1,3)*SA01(J,K2,1)*T(MP12)*
  2 SA01(L,K1,4)*SA01(J,K2,2)*T(MP13)*
  3 SA01(L,K1,3)*SA01(J,K2,2)*T(MP14)*
  F113,1,JI=F113,L,J1+SA(D(J,K1,2)*SA01(L,K2,3)*T(MP11)*
  1 SA01(L,K1,2)*SA01(J,K2,3)*T(MP12)*
  2 SA01(L,K1,1)*SA01(J,K2,4)*T(MP11)*
  3 SA01(L,K1,2)*SA01(J,K2,4)*T(MP14)*
  F114,1,JI=F114,L,J1+SA(D(J,K1,3)*SA01(L,K2,3)*T(MP11)*
  1 SA01(L,K1,4)*SA01(J,K2,3)*T(MP12)*
  2 SA01(L,K1,3)*SA01(J,K2,4)*T(MP11)*
  3 SA01(L,K1,4)*SA01(J,K2,4)*T(MP14)*
  121 CONTINUE
  120 CONTINUE
C INVERT F1 TO GET FINI
C
  150 CALL ESCIFI,N,MAXBK,F1
C SKIP FOR LAST BLOCK
C
  IF(N2.FEQ.0) GO TO 175
C CALCULATE -FINI*CIN+1 IN F1
C
  DO 160 L=1,NS
    DO 160 K=1,MAXJP
      IF(INPADJ(L,K).NE.0) GO TO 160
      J=NPA(JL,K)-N
      SADIL(L,K)=SADIL(L,K)+C0
    160 CONTINUE
C LOOP OVER NODES IN F1 (I AND J) AND COLUMNS OF F (ROWS OF C1 (L))
C
  CALL ERASFI+0.0*MAXBK*MAXBK,F1
  N6=0
  DO 160 L=1,NS
    DO 160 K=1,MAXJP
      IF(INPADJ(L,K).NE.0) GO TO 160
      J=NPA(JL,K)-N
      SADIL(L,K)=SADIL(L,K)+C0
    160 CONTINUE
C ADD NODE CONTRIBUTION TO F1
C
  DO 161 I=1,NS
    IF(WIN(I,L)) LI
    LF=MAXC(I,L)
    CALL MAT(F1,L,F1,L,2,2,TMP1,-1,1,F1(L,I,J))
    161 CONTINUE
    160 CONTINUE
C DELETE CIN+1 ENTRIES FROM ADJACENT STIFF TABLES
C
C LOOP OVER NODES (ROWS IN C1
C
  DO 160 I=1,NS
C CHECK AND ZERO OUT ENTRIES IN ADJACENT STIFF TABLES
C
  DO 180 K=1,MAXJP
    IF(INPADJ(L,K).LE.N21) GO TO 180
    180 M=1,4
    185 SADIL(K,M)=0.0
    NACJNP(I)=NACJNP(I);-
    NPADJ(L,K)=0
    180 CONTINUE
C SKIP FOR FIRST BLOCK
C
  175 IFINI.EQ.11 GO TO 205
C CONDENSE STIFFNESS TABLES
C
  DO 190 I=1,NS
    DO 195 K=1,MAXJP
      IF(INPADJ(L,K).NE.0) GO TO 195
      N2R=0
      DO 196 L=K,MAXJP
        IF(INPADJ(L,K).NE.0) GO TO 197
        N2R=N2R+1
        196 CONTINUE
        197 ISTT=K+N2R
        IF(ISTT.GT.MAXJP) GO TO 196
        DO 198 L=ISTT,MAXJP
          NPADJ(L,L)=NPADJ(L,L)
          DC(198,M)=1
        198 SADIL(L,M)=SADIL(L,M)
        DO 200 L=1,N7RO
          NPADJ(L,M)=NPADJ(L,L)
          200 M=1,6
        200 SADIL(L,M)=C0
      195 CONTINUE

```


३


```

C
C INITIALIZE
C
      AI(1,L,1)=T1
      AI(2,L,1)=T2
      AI(3,L,1)=T3
      AI(4,L,1)=T4
130 CONTINUEF
C
      DO 140 L=1,M
      DO 140 M=1,L,M
      AI(L,M)=AI(1,L,M)+AI(1,1,M)*TDF(1,1)
      AI(2,L,M)=AI(1,L,M)*TDF(1,2)
      AI(3,L,M)=AI(1,L,M)*TDF(1,3)
      AI(4,L,M)=AI(1,L,M)*TDF(1,4)
1     AI(2,L,M)=AI(1,2,L,M)+AI(2,1,M)*TDF(1,1)
      AI(3,L,M)=AI(1,2,L,M)*TDF(1,2)
      AI(4,L,M)=AI(1,2,L,M)*TDF(1,3)
      AI(1,L,4)=AI(1,3,L,M)+AI(1,2,M)*TDF(1,1)
      AI(2,L,4)=AI(1,3,L,M)*TDF(1,2)
      AI(3,L,4)=AI(1,3,L,M)*TDF(1,3)
      AI(4,L,4)=AI(1,3,L,M)*TDF(1,4)
3     AI(1,L,4)=AI(1,4,L,M)+AI(2,1,M)*TDF(1,1)
      AI(2,L,4)=AI(1,4,L,M)*TDF(1,2)
      AI(3,L,4)=AI(1,4,L,M)*TDF(1,3)
      AI(4,L,4)=AI(1,4,L,M)*TDF(1,4)
140 CONTINUEF
SON CONTINUEF
RETURN
END

SUBROUTINE LINK2A
IMPLICIT REAL(0.1A-H,0.7)
COMMON MAXP,MAXNP,MAXLINE,MAXLOAD,MAXFLA,NUMNP,NUMFL,
ISTRES,LINE$,$NUMBERL,LINE$NP($),PERIODNP,NUMLOAD(1,50),
1  RAN(1,50),ZANI(1,50),SNDR(1,50),TITLE(1,50),MAXED,
2  MBLCK,MBRDN,MAXMAX,TRFLAL,PDAMP,RTA,KPUN,
3  IPFLTP,KOUNT,TRFLLA,T,TMAX,TRFLUN
4  DIMNSION INP(12,1000),XN(2,1000),VFL(2,1000),ACL(2,1000),
15GMA(11,50),SIGMA(11,50)
DIMENSION NADJNP(25),ITVPF(35),TMFTA(35),XMS55(25),INMP(12,30),
INPADI(30,6),SADI(30,6,4),F(4,30,30),PMS(12,30),
DIMNSION STRFCS(4,6,50),NADJP(150),XLNAD(12,50),
15QINV(4,30,50),NUMARY(30)

INTGRATF TMF PARALEM AND WRITE THE DTW TAPF

```



```

525 CONTINUE
      CALL ERSE(2*N5,XNP1(I,N1))
C   DO FORWARD PART OF RECURSION FOR THIS PLANE
C
C   IF(N1,FQ,1) GO TO 560
C
C   DO 550 I=1,N7
      NUM=N0JNP1(I)
      IF(NUM,EQ,0) GO TO 550
      DO 555 KK=1,NUM
      JNP=APAD(J,KK)
      RHS1(I)=RHS1(I,1)-(SAND(I,KK,I)*XNP1(I,JNP)+  

      SAN(I,KK,2)*XNP1(I2,JNP))  

      RHS2(I)=RHS2(I,1)-SAD(I,KK,3)*XNP1(I,JNP)+  

      SAD(I,KK,4)*XNP1(I2,JNP))
      555 CONTINUE
      550 CONTINUE
C
C   560 EN 565 I=1,N5
      EN 565 J=1,N5
      (F=MINT(I,J))
      JF=MAXN(I,J)
      CALL MM1(F(1,IF,JF),2,Z1,RHS1(I,J),J-1,I,XNP1(I,I+N1-1))
      IF(N2,LT,NUMNP1) GO TO 560
C
C   DO BACKWARD PART OF RECURSION
C
C   600 IF(ICP,EQ,1) GO TO 615
      READ(2,DININEXT,N5,N6,((F(I,J,K),I=1,4),J=1,N5),K=1,N6)
      NI=NINEXT+N5
      DO 610 I=1,N5
      DO 610 J=1,N6
      CALL MM1(F(1,1,J),2,Z1,XNP1(I,1,J+N1-1),I,1,XNP1(I,I+N1,NF))
      610 ININEXT,GT,1) GO TO 600
      615 REMIND 20
C
C   CALCULATE REST OF DIFFERENCE SCHEME
C
C   DO 325 I=1,NUMNP
      ACLU=ACL(I,I)
      ACLU=(XNP1(I,1)-XN(I,1))-VEL(I,1)*DT+(BFTA-I)*DT*DT*ACL(I,I)
      1/(BETA*DT*DT)
      ACLW=ACL(2,I)
      ACL(2,I)=XNP1(2,I)-XN(2,I)*DT+(BFTA-I)*DT*DT*ACL(2,I)
      1/BETA*DT*DT
      VEL(I,1)=VEL(I,1)+0.5*NT*ACLU+ACL(I,I)
      VEL(2,I)=VEL(2,I)+0.5*NT*(ACLU+ACL(2,I))
      XN(1,I)=XNP1(1,I)
      XN(2,I)=XNP1(2,I)
      325 CONTINUE
      COUNT=COUNT+1
      RFAC(2,I)
      GO TO 100
      END FILE 8
      RFWINC 8

```

```

RFWIND 20
RETURN
END

```

```

      SUBROUTINE ASTRESS(MXL(INF,MXLAD),DAONP,L(NFS,RAD,ZAD,SMP))
      ISIGMAU,SIGMAN,T,TX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LOADNP(MXLINF),PADIMXL(INF,MXLAD),ZAD(MXLINE,MXLAD),
      ISNORM(MXLINE,MXLAD),SIGMAU(MXLNF,MXLAD),SIGMAN(MXLNF,MXLAD)
      DATA IRD/0/
C
C   CALCULATE BOUNDARY STRESSES
C
      IQT=IRD+I
      TK=C*D
      NUM=LOADNP(I)
      DO 10 I=1,NUM
      SIGMAU(I,I)=0.0
      10 SIGMAN(I,I)=0.0
      IF(IRD-2)107,200,300
C
C   100 REACT(S,IND0,IPULSE,IOIREC,N1,N2,STAMP
      100 FORMAT(4,15,F10.2)
C
C   IPULSE = 0 DELTA FUNCTION STRESS
C   = 1 STEP FUNCTION STRESS
C   IDIREC = 0 STRESS IN TRANSVERSE DIRECTION
C   = 1 STRESS IN NORMAL DIRECTION
C   N1      FIRST LOADED NODE POINT
C   N2      LAST LOADED NODE POINT
C   STAMP   STRESS AMPLITUDE
C
      IF(N1,FEQ,0) N1=1
      IF(N2,FEQ,0) N2=LOADNP(I)
      IF(STAMP,FEQ,0.0) STAMP=F1.0
      WRITE(6,2000)IPULSE,IOIREC,N1,N2,STAMP
      2000 FORMAT(9H,IPULSE=,15,1OH, IDIREC= ,15,6H, N1= ,15,6H, N2= ,15,
      19H, STAMP= ,F10.2)
      RETURN
C
C   200 IF(I,NE,2,0,AND,IPULSE,EQ,0) RETURN
      205 DO 210 I=N1,N2
      CS=DCNS(SNORM(I,1))
      SN=DSNS(SNORM(I,1))
      IF(IDIREC,NE,0) GO TO 206
      SIGMAU(I,I)=-STAMP*SN
      SIGMAN(I,I)=STAMP*CS
      GO TO 210
      206 SIGMAU(I,I)=STAMP*CS
      SIGMAN(I,I)=STAMP*SN
      210 CONTINUE
      RETURN
C
C   300 IF(IPULSE,FEQ,1) GO TO 295
      RETURN

```

END.

```
C C POSITION STRFSS TAFF PAST TITLE RECORD
C C
C SUBROUTINE LINK3
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON MAXNP,MXADJP,MXLINP,MLXLP,MAXFLR,NUMNP,NUMFL,
C 1 ISTRES,LINES,NUMPFL,LOADNP(1),PERIOD,DT,TITLE(1C),MAXBN,
C 2 RAD(1,5D),ZAD(1,5D),SNORM(1,5D),TITLE(1C),MAXBN,
C 3 MALOCK,NREADS,MAXBK,TREAL,PDAWP,RETA,KRUN,
C 4 NUMDUT,TMAX,TSRSS,JDUT,JDOUT
C DIMENSION UDNN(1CDU),UDDN(100C),UDNN(100C),UDNN(100C),
C 1WN(1DDO),PRESSU(1,100),PRESSW(1,100),FILLA(200)
C DIMENSION STNPW(4,100),STNPW(4,100),STANU(4,100,8)
C 1,NADJEL(100C),NADJNP(100C),NPDU(100C),NPNU(100C)
C DIMENSION BLOCKA(3200),BLOCKB(3200)
C EQUIVALENCE UDNN(1),BLOCKA(1),(UDNN(1)),BLOCKA(101),
C 1,BLOCKA(2001),(PRESSU(1)),BLOCKA(3001),(PRESSW(1,1),BLOCKA(3101),
C 2),(WN(1),BLOCKB(1)),(WN(1),BLOCKB(1001)),(WN(1),BLOCKB(2001)),
C 3,(FILLA(1),BLOCKB(3001))
C DIMENSION COM(198)
C EQUIVALENCE(MAXNP,COM11).
C
C THIS LINK PROCESSES THE 0TH TAPE AND CALCULATES STRESSFS.
C PRINTS AND WRITES BOTH ON THE DTM TAPE
C
C REWIND 8
C POSITION 0TH TAPE PAST TITLE AND COMMON BLOCKS
C
C READ(BTITLE
C READ(8)CDM
C CALL LINK3A(BLOCKA,BLOCKB,STNPW,STNPU,STADW,STADU,NADJEL,
C 1NPADJ,NPDU,NPTP)
C CALL LINK3B(UDNN,WDRN,WDN,WN,WN,PRESSU,PRFSSW,STNPU,STNPP,
C 1STADU,STADW,NADJEL,NADJEL,NADJEL,NADJEL)
C
C RETURN
C END
C
C SUBROUTINE LINK3A(STADU,TTADW,STNPU,STNPW,STADU,STADW,NADJEL,
C 1NADJNP,NPDU,NPTP)
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON MAXNP,MXADJP,MXLINP,MLXLP,MAXFLR,NUMNP,NUMFL,
C 1 ISTRES,LINES,NUMPFL,LOADNP(1),PERIOD,DT,NPLDN(1,5C),
C 2 RAD(1,5D),ZAD(1,5D),SNORM(1,5D),TITLE(1C),MAXRD,
C 3 MALOCK,NREADS,MAXBK,TREAL,PDAWP,RETA,KRUN,
C 4 NUMDUT,TMAX,TSRSS,JDUT,JDOUT
C DIMENSION STNPW(4,100),STNPW(4,100),STADU(4,100,8),STADW(4,100,8)
C 1,NADJEL(10C),NADJNP(100),NPADJ(100,8),NPDU(100),NPNU(100)
C DIMENSION NPTN(100),ANAF(6),MADJNP(100),MADJFL(100),
C 1MPADJ(100,8),TTNPW(4,100),TTADU(4,100,8),
C 2TTADW(4,100,8)
C
C READ CARD INPUT AND STRESS TAUFLS OFF BLOCKFD STIFF. & STRESS TAPE
```

A-35

51

TFC(KRUN,FQ,3).OP.(KRUN,FQ,4) GO TO 100

```

C 7 FOR CUT STRESS TAPE
C
C NO 11 I=1,100
C NADJNP(I,I)=0
C WADJNP(I,I)=0
C NADJEL(I,I)=0
C WADJEL(I,I)=0
C DO 12 J=1,9
C   NPANJ(I,J)=0
C 12 MPADJ(I,J)=0
C DO 11 K=1,4
C   STNPK(I,K)=0.0
C   TTNPK(I,K)=0.0
C   STNPW(I,K)=0.0
C   TTNPW(I,K)=0.0
C   RO 11 J=1,9
C   STADUK(I,J)=0.0
C   TYANUK(I,J)=0.0
C   STAUK(I,J)=0.0
C   TTADUK(I,J)=0.0
C 11 CONTINUE
C
C READ STRSS TABLES FOR OUTPUT NODFS
C
C NODF=0
C READ(1,*) NPL0,NPHIGH,
C      NUNMNP,(WANJNP(I,J),I=1,4),J=1,MXANJNP
C      TTNPK(I,J),I=1,4),J=1,MXANJNP,I=1,MUXNP
C      NP0=NPL0-1
C
C DO 13 I=1,NUMNUT
C   NP0=NPL0+1
C   IMPAC(I,J)=WANJNP(I,J)
C   SEL(NPNEW,I,T,NPL0).NP=(NPNEW.GT.NPHIGH) GO TO 13
C   NOUT=NUMNUT+1
C   NP=NPL0-NP0FF
C   NADJNP(I)=WADJNP(NP)
C   NADJEL(I)=WADJEL(NP)
C   DO 14 J=1,MXADJ
C     NPANJ(I,J)=WP ADJ(NP,J)
C   DO 15 K=1,6
C     STNPK(I,J)=TTNPW(I,K,NP)
C     STNPW(I,K)=TTNPW(I,K,NP)
C     DO 16 J=1,MXADJ
C       STADUK(I,J)=TYANUK(NP,J)
C       STAUK(I,J)=TYANUK(NP,J)
C 14 CONTINUE
C 15 CONTINUE
C 13 CONTINUE
C
C IF(NMUT.GT.NMUT) GO TO 17
C IF(NPHIGH.LT.NUMNP) GO TO 16
C WRITE(6,17) NMUT,NMUT
C 17 FORMAT(1H1,14HF,F9.6) NMUT=15,THUMNUT=15/15,THANUT = 15
C CALL EXIT
C
C 12 RFWINC 1

```

RETURN
END

```

SUBROUTINE LINK2(UNION,WPN,WN,UN,WN,PRESSU,PRESSW,STNP1).
1$NPW,STADU,STAUW,NADJL,NADJNP,NADJ,INPUT,NPTP1
IMPLICIT REAL(8)A-H=0.1
COMMON MNP,MXACUP,MXLINP,MXNPD,MXNPA,MAXFL,NUMFL,NUMEL,
1$TRES,LINES,NUMPFL,LMNPFL(1),PERIOD,DT,NPLN(I,5),I=1,4/72
2$RAN(1,5),ZD(1,5),SGRM(1,5),TITLE(1),MAXRN,
3$WALCK,NPFANS,MAXBCK,TRFLAD,DOAMP,RETA,KRUN,
4$NUMNUT,TWAK,TSTRSS,INUT,JCUT
DIMENSION UNDN(100),WNDN(100),UN(100),
1$WN(100),SIGMAW(1,5),SIGMAW(1,5),
2$NADJEL(1,4),NADJNP(1,4),NPANJNP(1,4),NPCTP(1,4),
3$MFNSTON SIG(4),NODFFL(24),NP1?26*4),EPSD(26*4),ISIGPL(100,4),RUFF(160)
C CALCULATE STRESSSES AND WRITE THF NTH TAPE
C
C WRITE(5,1) TITLE RECORD ON NTH TAPE
C
C RFWINC 3
C WRITE(3) TITLE
C
C INITIALIZE AND WRITE FIRST RECORD ON 0TH TAPE
C
C COUNT=1
C NOWN=1
C JC=JCUT
C IC=ICUT
C FT=0.0FL0AT(JCUT)
C NWNS=16*NMUT
C WRITE(3,1) TAUX,FT,TRFL,NUMNUT,INPUT(I),I=1,NMUT
C
C DO 2 I=1,4
C   SIG13=0.0
C   SIGMX=0.0
C   SIGMN=0.0
C   THFT=0.0
C 2 POSITION NTH TAPE AND PRINT START TIME
C
C IF(STRESS.GT.(0.5,0)) GO TO 3
C T=-NT
C GO TO 7
C CONTINUE
C IF(NREADS.EQ.1) GO TO 4
C
C NMUT=NPFANS-1
C DO 5 I=1,NUMFL
C   READ(1,NUMFL
C   READ(1,RT
C   IF(STRESS-T).GT.(1.5,0) GO TO 1
C   TM=TAPEFL+DT
C   TSTAR=T+DT
C   WRITE(6,4) TSTAR,TW
C
C Reproduced
C best available from
C copy.

```

```

6 FORMAT//35H OUTPUT HISTORY TAPF STARTS AT TIME
  (IPF15.5,5 SEC./36H WHICH CORRESPONDS TO A REAL TIME OF.
21PF15.5,5 SEC.
C ISWTCM=1
C N=11
202 SIG(J)=SIG(J)+SIG(I,J,K)*FPS(I,J,K);
C
C MAIN LOOP FOR EACH TIME STEP ON DTH TAPE STARTS HERE
C
C 400 CONTINUE
C INCREASE COUNTERS, CHFCR, AND SKIP A TIME STEP ON DTH TAPE
C IF NECESSARY
C
C IC=IC+1
JC=JC+1
IF(11C.GT.JOUT).OR.(JC.GT.JMINT) GO TO 401
DO 402 I=1,NREADS
402 REAC(8,END=999,FRR=999)NUMELA
GO TO 400
C SKIP FOR SUMMARY OUTPUT
C 401 IF(IKRUN.EQ.31).OR.(IKRUN.EQ.41) GO TO 100
C ZERO OUT PLASTIC STRESS ARRAY
C
C DO A 1=1,NUMOUT
DO A J=1,4
  A SIGPL(I,J)=0.0
C SKIP FOR NO PLASTIC ELEMENTS
C IF(1NUMPFL.EQ.0) GO TO 300
C CALCULATE PLASTIC STRESS TABLES
C
C NUMCFL=0
209 REAC(8,END=999)NUMFLA,(NODNEFL11),(NP((1,J),EPS(1,J),
11C(I,J,K),K=1,4),J=1,4),I=1,NUMELA
C
C DO 200 I=1,NUMEL
DO 200 I=1,NUMEL
  I=I
  JJ=0
  KK=0
  LL=0
C 201 CONTINUE
C
C DO 201 J=1,NUMOUT
  I(NPTP1(J)).EQ.NP(I,1) II=J
  I(NPTP1(J)).EQ.NP(I,2) JJ=J
  I(NPTP1(J)).EQ.NP(I,3) KK=J
  I(NPTP1(J)).EQ.NP(I,4) LL=J
C
C 202 J=1,4
NM=MAX((1,J,K,LL)
1F(MM,FO,0) GO TO 204
C
C 203 SIGPL(N,J)=SIGL(N,J)+SIG(J)
204 GO TO (206,207,208,200),ISWTCM
C
C 205 ISWTCM=2
N=JJ
GO TO 205
207 ISWTCM=3
N=KK
GO TO 205
208 ISWTCM=4
N=LL
GO TO 205
C
C 209 (CONTINUE
C
C NM=NUMCFL,LT,NUMPEL) GO TO 209
  I=(NUMCFL,LT,NUMPEL)
DC 210 I=1,NUMOUT
DUM=NADJEL(I)
DC 210 J=1,4
  210 SIGPL(I,J)=SIGPL(I,J)/DUM
GO TO 100
C SKIP PAST PLASTIC DATA ON DTH TAPE
C
C 100 IF(NREADS.FO,1) GO TO 300
  I=NUM=NREADS-1
  DO 101 I=1,10UM
    101 READ(8,END=999,FRR=999)NUMELA
C
C READ A TIME STEP OFF DTH TAPE
C
C 300 10UM=LNAONDPL()
  RFA0( A,END=999,ERR=999)LT,KOUNT,TM,(UN((1,UNNN((1,MNC((1,
  1WN((1,WDNN((1,(I=1,NUMNP),SIGMAUT,(1,SIGMAUT,(1,I=1,TRUN))
C
C SKIP FOR NO PRINTED OUTPUT
C
C 350 FORMAT(1H1,5H(ME=,1P15.5H SF0...5X,6HKOUNT=,15,5X,
  11CHRFAL TME=,1P15.5H SF0...//)
  (FLIKRIN,FO,21,0P,TRUN,FO,41) GO TO 352
  WRITE(6,253)
C
C 352 FORMAT(2X,4HNOTE,4X,12HN(SP, U ((1,2X,12HVFL,UN ((PS),2X,

```

```

PW=SIGMAW(J,K)
GO TO 310
CONTINUE
  PU=C.0
  PW=D.0
C   CONVRT ACCELERATIONS TO G'S
C   ACCFLU=UDDN(NPNEW)/386.4
  ACCELM=WDN(NPNEW)/3R6.4
C   SKIP FOR NO PRINTFO OUTPUT
C   IF(JC,LF,INUT) GO TO 311
C   PRINT OUTPUT (COMPLFTE OR SUMMARY)
C   IF(IKRUN,FQ,3).OR.(IKRUN,FQ,4) GO TO 355
  WRITE(6,356) NPOUT(J),UN(NPNEW),UDN(NPNEW),ACCFLU,PU,SIG(1),
  LSIG(3),SIGMX,THFTA,WN(NPNEW),WDN(NPNEW),SIG(4)
  2SIGMN
  356 FORMAT(15,3X,1PAF14.4/BX,1P7F14.4//)
  GO TO 311
  355 WRITE(6,357) NPOUT(J),UN(NPNEW),UDN(NPNEW),ACCFLU,PU,WN(NPNEW),
  1WN(NPNEW),ACCFLU,PM
  357 FORMAT(15,3X,1P8F14.4)
C   LOAD BUFFER FOR TAPF OUTPUT
C   311 IF(JC,LF,JOUT) GO TO 301
  BUFFI IK,1=NPOUT(J)
  BUFFI IK+1=UN(NPNEW)
  BUFFI IK+2=UDN(NPNEW)
  BUFFI IK+3=ACCFLU
  BUFFI IK+4=PU
  BUFFI IK+5=SIG(1)
  BUFFI IK+6=SIG(3)
  BUFFI IK+7=SIGMX
  BUFFI IK+8=THETA
  BUFFI IK+9=WN(NPNEW)
  BUFFI IK+10=WDN(NPNEW)
  BUFFI IK+11=ACCFLU
  BUFFI IK+12=PM
  BUFFI IK+13=SIG(2)
  BUFFI IK+14=SIG(4)
  BUFFI IK+15=SIGN
  IK=IK+16

C   WRITF TAPE IF TAPF OUTPUT THIS STFP
C   301 CONTINUE
  IF(JC,LF,JOUT) GO TO 473
  WRITE(3,473)
  473 T,(BUFF(IK),K=1,NWDS)

C   SFT SWITCHFS, LOOP BACK, OR WRAP UP
C   GO TO 310
  212HSIGMAZ (PSI),2X,12HSIGMAR (PSI),2X
  212HSIGMAZ (PSI),2X,12HSIGMX (PSI),3X,114THFTA (DEGI)/1CX,
  312HDISP, W (IN),2X,12HVEL*WD (PSI),2X,12HACCL, WDD (GI),2X,
  412HSIGMAW (PSI),2X,12HSIGMAT (PSI),2X,12HACCL (PSI),2X
  512HSIGMN (PSI),2X
  GO TO 351
  352 WRITF16,354
  354 FORMAT(12X,4MD0F,6X,12HDISP, U (IN),2X,12HVEL, UD (IPSI),2X,
  112HACCL, WDD (GI),2X,12HSIGMAW (PSI),2X,12HDISP, W (IN),2X,
  212HVEL, WD (IPSI),2X,12HACCL, WDD (GI),2X,12HSIGMAW (PSI),2X
C   SFT-UP OUTPUT FOR EACH OUTPUT NODE
C   351 IK=1
  DO 301 I=1,NUMNUT
  NPNEW=NPTP(I)
C   SKIP STRFSS CALCULATION FOR SUMMARY OUTPUT
C   IF((IKRUN,FQ,3).OR.(IKRUN,FQ,4)) GO TO 321
  CALCULATF STRFSSFS
  NUM=NAJNP(1)
  DO 302 J=1,4
  SIG(J)=STPUI(J,I)*UN(NPNEW)+STNPW(J,I)*WN(NPFW)
  DO 3D3 K=1,NUM
  NODE=NPADJ(I,K)
  SIG(J)=SIG(J)+STADU(J,I,K)*UN(NODF)+STADW(J,I,K)*WN(NODF)
  303 SIG(J)=SIG(J)-SIGL(I,J)
  302 CONTNUF
C   CALCULATF PRINCIPAL STRFSSFS AND ANGLF
C   DUM1=1SIG(1)-SIG(3))/2.
  DUM2=DUM1*DUM1+SIG(4)*SIG(4)
  IF(DUM2>GT.C.0) GO TO 321
  RADIUS=0.0
  GC TO 322
  321 RADIUS=DSORT(DUM2)
  DUM3=1SIG(1)+SIG(3))/2.
  FIG1=DUM3*RADUIS
  EIG2=DUM3*RADIUS
  SIGMX=FIG2
  IF(DARS(FIG1)*GT.DABS(FIG2)) SIGMX=FIG1
  SIGMN=FIG1*FIG2-SIGMX
  CALL ARCTAN(SIGMX-SIG(3)*SIG(4),THFTA)
C   SFT-UP BOUNDARY STRFSSFS
C   301 DO 3C9 J=1,LINFS
  NUM=LOAJP(J)
  DO 309 K=1,NUM
  NODE=NPLNAN(J,K)
  IF((NODEF=NPTP(I))) GO TO 309
  PI=SIGMAU(J,K)

```

```

403 IF (IC.GT.10UT) IC=1
1F(JC.GT.JDUT) JC=1
GO TO 400
990 CONTINUE
C
END FILE 3
REHIND 3
RETURN
END

```

```

SUBROUTINE ARCTAN(X,Y,THETA)
IMPLICIT REAL*8(A-H,D-Z)
DATA PI/3.14159265DC/
C
C GET ARCTANGENT IN CORRECT QUADRANT BETWEEN
C +PI AND -PI
C
IF (X)50,10,50
10 IF(Y)40,30,20
20 THETA=PI/2.0
RETURN
30 THETA=0.0D0
RETURN
40 THETA=-PI/2.0
RETURN
50 THETA=DATAN(DASIN(Y/X))
IF (X)90,60,60
60 IF(Y)70,80,80
70 THETA=-THETA
80 RETURN
90 IF(Y)110,100,100
100 THETA=PI-THETA
RETURN
110 THETA=-PI+THETA
RETURN
END

```

Reproduced from copy.
 Best available

APPENDIX A1. (MINBND) SOURCE LISTING.

```

SUBROUTINE MINBND(IRON,ICOL,NUMN,IS,JS,MBN)
DIMENSION IRON(1),ICOL(1),NUMN(1),IS(1),JS(1)
C MATRIX BANDWIDTH MINIMIZATION - - KUSEN 11968
C
      NCP=0
      RESTIND=8
      DO 20 I=1,NUMN
      NBD=181*(I-1)+MAX(K,J-1)*MAX(K,J-1)
      TH=20 J=1,NAI
      I=I-NAI J=J-2D,J0
      10 NCP=NCP+1
      ICOL(1)=CP+1
      INDF(1)=INDF(1)+1
      20 CONTINUE
      RESTIND=8
      CALL NUMN(1,INDF,ICOL,IS,NCP,NUMN,MBN)
      DU 30 I=1,NUMN
      30 J=I+1
      RETURN
      END

SUBROUTINE NUMBERET(IRON,ICOL,IS,NCP,NUMN,MAXBND)
DIMENSION IRON(1),ICOL(1),IS(1),MAXBND
MBDPRV=NSIZE
MAXPM2=NSIZE
1 MAXBND=0
DO 2 I=1,NCP
JULF=IRON(I)-ICOL(I)
IF(JDIFF-MAXBND)>2,1000
1000 MAXBND=JDIFF
IRMXBD=IRON(I)
ICMXBD=ICOL(I)
2 CONTINUE
PRINT 6000, MAXBND
6000 FORMAT(6W MAXBANDWIDTH = .15)
IF(MAXBND-1)>100,1004,1001
1001 ISSTART=XMAXUF(1,(2*ICMXBD-1)KMXBD+1)
ISTUP=IRMXBD-1
JJ=IRMXBD
DU 3 I=START,ISTUP
3 CONTINUE
ISSTART=ICMXBD+1
ISTUP=MINUT(NSIZE,(2*ICMXBD-1))
CALL SEARCH(1,NUMN,ICOL,IRMXBD,I,NCP,NUFUND,MAXBND)
1002 I=I
JJ=ICMXBD
DU 4 I=START,ISTUP
4 CONTINUE
IF(I-IRMXBD)>1003,4,1003
1003 I=I
CALL SEARCH(1,NUMN,ICOL,IRMXBD,I,NCP,NUFUND,MAXBND)
IF(NUFUND-1)>0,4
4 CONTINUE
I022 IF(MAXBND-MBXPKV)>100,7,1010
1010 I=I
DU 8 I=I
8 CONTINUE
NIPMS2=0
      GO TO 9
      IF(INTPHM2>19,1004,9)
      9 RETURN
      CONTINUE
      N2INT=2*INTPHS2
      NPHS2=INTPHS2+1
      ISTART=XMAXUF(1,(2*ICMXBD-1)KMXBD)
      ISTUP=IRMXBD-1
      JJ=IRMXBD
      DO 31 I=START,ISTUP
      IFL=ICMXBD+1
      ISTOP=XMINOF(NSIZE,(2*ICMXBD-1)KMXBD)
      1005 I=I
      CALL P2SRCH(IRDW,ICUL,IRMXBD,I,NCP,NUFUND,MAXBND,INTPH2,N2INT)
      IF(INFOUND-1)>131,6,31
      31 CONTINUE
      ISTART=ICMXBD+1
      ISTOP=XMINOF(NSIZE,(2*ICMXBD-1)KMXBD)
      JJ=ICMXBD
      DU 44 I=START,ISTUP
      IFL=IRMXBD+1
      ISTOP=IRMXBD+1
      1006 I=I
      CALL P2SRCH(IRDW,ICUL,IRMXBD,I,NCP,NUFUND,MAXBND,INTPH2,N2INT)
      IF(INFOUND-1)>4,6,44
      44 CONTINUE
      RETURN
      6 CALL INTACH(IRDW,ICUL,JJ,I,I,NCP)
      M1=JJ
      M2=II
      DO 37 I=1,NSIZE
      IFL(S(I))-M1)1008,1007,1008
      1007 J=I
      1008 IFL(S(I))-M2)137,1009,37
      37 CONTINUE
      IFL(J1)=M2
      IFL(J2)=M1
      GU TO 1
      END
      SUBROUTINE INTACH(IRDW,ICUL,I,J,I,NCP)
      DIMENSION IRD(1),ICUL(1)
      DU 1 I=1,NCP
      IFL(IRD(I))-IIS)2,5000,2
      5000 CALL SWITC(IRD(I),ICUL(I),I,NCP,J,JL)
      IROW(I)=JH
      ICQ(I)=JL
      GU TO 1
      2 IFL(IRD(I))-IIS)3,5001,3
      5001 CALL SWITC(IRD(I),ICUL(I),I,NCP,J,JL)
      IROW(I)=JH
      ICQ(I)=JL
      GU TO 1
      3 IFL(ICUL(I))-IIS)4,5002,4
      5002 CALL SWITC(ICUL(I),I,NCP,J,JL)
      IROW(I)=JH
      ICQ(I)=JL
      GU TO 1
      4 IFL(ICUL(I))-IIS)5,5003,1
      5003 CALL SWITC(ICUL(I),I,NCP,J,JL)
      IROW(I)=JH
      ICQ(I)=JL
      GU TO 1
      56
      
```

```

ICOL(1)=JL
1 CONTINUE
RETURN
END
SUBROUTINE PSEARCH(ROW,ICOL,IC,JC,MCP,NF,MAXBND,INTPH2,M2INT)
DIMENSION IR0M(1),ICUL(1),INTPH2,M2INT
NF=0
DO 1 I=1,MCP
  IF(IROW(1)-JC12+4000,2
  4000 ICAN2=XABSF1JC-ICQ(111
  IF(ICAN2-MAXBND)1,1,4001
  4001 RETURN
  2 IF(ICOL(1)-JC13+4002,3
  4002 MCN2=XABSF1JC-IR0M(111
  IF(MCN2-MAXBND)1,1,4001
  3 IF(IROW(1)-JC14+4003,4
  4003 ID02=XABSF1JC-ICD(111
  IF(ID02-MAXBND)1,1,4001
  4 IF(ICOL(1)-JC11+4004,1
  4004 M0D02=XABSF1JC-IR0M(111
  IF(M0D02-MAXBND)1,1,4001
  1 CONTINUE
  1 IF(M2INT)4005,5,4005
  4005 M11=0
  M1J=0
  DJ=1+1*M2INT
  1 IF(INTPH2(1)-IC161,4006,61
  4006 M11=1
  61 IF(INTPH2(1)-JC16,4007,6
  4007 M1J=1
  6 CONTINUE
  4008 IF(M11+M1J-215,4001,5
  5 CONTINUE
  NF=1
  INTPH2(M2INT+1)+JC
  INTPH2(M2INT+2)+JC
RETURN
END
SUBROUTINE SEARCH(ROW,ICUL,IC,JC,MCP,NF,MAXBND)
DIMENSION IR0M(1),ICOL(1)
NF=0
DO 1 I=1,MCP
  IF(IROW(1)-JC12+3000,2
  3000 MCAN=XABSF1JC-ICQ(111
  IF(MCAN-MAXBND)1,3001,3001
  3001 RETURN
  2 IF(ICOL(1)-JC13+3002,3
  3002 MCAN=XABSF1JC-IR0M(111
  IF(MCAN-MAXBND)1,3001,3001
  3 IF(IROW(1)-JC14+3003,4
  3003 IDU=XABSF1JC-ICU(111
  IF(IDU-MAXBND)1,3001,3001
  4 IF(ICUL(1)-JC11+3004,1
  3004 MUDU=XABSF1JC-IR0M(111
  1 IF(MDU-MAXBND)1,3001,3001
  1 CONTINUE
  NF=1
RETURN
END
SUBROUTINE SEARCH(ROW,ICUL,IC,JC,MCP,NF,MAXBND)

```

APPENDIX B. OS/360 SLAMCODE LINK EDITOR CONTROL CARDS.

INSERT MAIN
OVERLAY ALPHA
INSERT LINK1
OVERLAY BETA
INSERT LINK1A, LOADIT, LNK1A, UNLINK, VAL, J, MATH, M1, GSJK1, SUKT1
OVERLAY META
INSERT LINK1B, ELAST, STIFF, INS, DISP, ADJUST, MASS, PRNK
OVERLAY BETA
INSERT LINK1C, STRESS, PDIS, AVG
OVERLAY BETA
INSERT LINK1D, SIZE, CLSTER
OVERLAY BETA
INSERT LINK1F, ELUST
OVERLAY ALPHA
INSERT LINK2, MM1, ERASE
OVERLAY GAMMA
INSERT LINK2A, DISKE, ESC
OVERLAY GAMMA
INSERT LINK2B, HSTRES
OVERLAY ALPHA
INSERT LINK3
OVERLAY KHU
INSERT LINK3A
OVERLAY KHU
INSERT LINK3B, AKCTAN
ENDY MAIN

1/3

576

APPENDIX C. OS/360 FORTRAN H SLAMCODE JOB CONTROL LANGUAGE CARDS.

a.) Requires two 9-track tape drives.

```

//>L(8),
//. EXEC PNL=SYSLIB(LIBECL)=LIST,PAR=19LY*
//SUBCT=INPUT(0,0,1)=2400,VOL=SYSDU,BSIZE=1200,DSIZE=51200
//DATA,FTU1F001 00 UNIT=5YS01,SPACE=(364,125) ,
X
  IC0=RECFL=Y05,LRECL=37,BLKSIZE=3641
//DATA,FTU3F001 00 UNIT=1SYS01,SECFL=01,001,BSIZE=(19L,10),
X
  DSPP=1NETH,ASS1,DSIZE=1011,
X
  IC0=(RECFL=Y5,LRECL=364,BSIZE=3408)
//DATA,FTU4F001 00 UNIT=1SYS01A,SPR=(FTU1F001,FTU3F001),
X
  SPACE=(364,100),
X
  IC0=(RECFL=Y6,LRECL=36,BSIZE=364)
//DATA,FTU6F001 00 UNIT=1SYS01A,SECFL=(FTU1F001),
X
  SPACE=(484,100),UCB=(RECFL=Y65,LRECL=46,BSIZE=404)
//DATA,FT10F001 00 UNIT=2400,VIL=52W=0123H,USIZE=511F1,Label=1,
X
  IC0=(RECFL=Y5,LRECL=2052,BLKSL=205)
//DATA,FT12F001 00 UNIT=1SYS01,SPR=(FTU3F001,FTU4F001),
X
  SPACE=(1804,125)+0Ln=(RECFL=Y5,LRECL=160,BSIZE=160)
//DATA,FT14F001 00 Objcty
X
  IC0=(RECFL=Y5,LRECL=201,BSIZE=660)Xpart,
//DATA,FT22F001 00 UNIT=5YS01,SPACE=(19L,20),
X
  IC0=(RECFL=Y5,LRECL=2052,BSIZE=2050)
//DATA,FT24F001 00 UNIT=19L,20) ,
X

```

Reproduced from
less available copy.

A-1.4

APPENDIX C. OS/360 FORTRAN HSLANCODE JOB CONTROL LANGUAGE CARDS.

b.) Requires four 9-track tape drives.

A-1

Reproduced from
best available copy.